数据科学与大数据技术系列

Python 数据挖掘方法及应用

王斌会　王　术　编著

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书重点介绍 Python 语言在数据处理与数据挖掘方面的应用技巧，主要包括数据分析基础知识（数据收集与分析软件、数据挖掘的分析基础、简单数据的统计分析），数据分析高级方法（多元数据的综合分析、时序数据的模型分析），大数据基本处理方法（大数据分析基础应用、文献计量与科研评价、社会网络分析方法、数据分析编程平台）等内容。附录中还提供了 Python 数据分析相关方法和函数等，方便读者随时查看。本书内容丰富，图文并茂，可操作性强且便于查阅，主要面向数据分析的读者，能有效帮助读者提高数据处理与分析的水平，提升工作效率。书中的例子数据、习题数据及相关代码都可在作者的学习博客 http://blog.leanote.com/DaPy 下载使用，也可登录华信教育资源网 http://www.hxedu.com.cn 免费下载。

本书适合各层次的数据分析用户，既可作为初学者的入门指南，又可作为中高级用户的参考手册，同时也可作为各大中专院校和培训班的数据分析教材。

图书在版编目(CIP)数据

Python 数据挖掘方法及应用 / 王斌会，王术编著. —北京：电子工业出版社，2019.3
（数据科学与大数据技术系列）
ISBN 978-7-121-34495-4

Ⅰ. ①P… Ⅱ. ①王… ②王… Ⅲ. ①软件工具－程序设计－高等学校－教材 Ⅳ. ①TP311.561

中国版本图书馆 CIP 数据核字(2018)第 125853 号

策划编辑：秦淑灵
责任编辑：秦淑灵
印　　刷：
装　　订：
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编：100036
开　　本：787×1092　1/16　印张：13.5　字数：340 千字
版　　次：2019 年 3 月第 1 版
印　　次：2019 年 3 月第 1 次印刷
定　　价：49.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010)88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：qinshl@phei.com.cn。

前　言

人类从农耕社会进入工业社会用了上千年时间，从工业社会进入信息社会用了一百多年时间，而从信息时代进入数据时代仅用了不到十年时间。随着互联网、物联网、云计算的不断深入应用，产生了大量的数据，这些数据的挖掘和分析应用，需要人们掌握数据分析技术。人类正全面进入大数据分析时代。

需要是发明之母。近年来，数据挖掘引起了信息产业界的极大关注，其主要原因是，存在大量的数据，可以被广泛使用，并且迫切需要将这些数据转换成有用的信息和知识。获取的信息和知识可以应用于各种领域，包括商务管理、生产控制、市场分析、工程设计和科学探索等。

“人生苦短，我要用 Python”，这是网上对 Python 评价最多的一句话，说明 Python 作为一种新兴的编程语言，已深入人心。现在我国许多地区高考试卷中都加入了 Python 编程的内容，一些中小学也开始开设 Python 编程课程。

本书重点介绍 Python 语言在数据处理与数据分析方面的应用技巧，涉及数据的整理、数据的输入和输出、探索性数据分析、基本数据分析、多元数据分析、时间序列数据分析、网络爬虫技术、社会网络分析、知识图谱和文献计量研究等数据分析方面的内容。附录中还提供了 Python 数据分析相关方法和函数等，方便读者随时查看。

全书分三部分，共 9 章内容。第一部分主要讲解数据分析基础知识，包括第 1、2、3 章，重点介绍数据收集与分析软件、数据挖掘的分析基础，以及简单数据的统计分析；第二部分讲解数据分析高级方法，包括第 4、5 章，主要介绍多元数据的综合分析和时序数据的模型分析；第三部分讲解大数据基本处理方法，包括第 6、7、8 章，重点介绍大数据分析基础应用、文献计量与科研评价、社会网络分析方法和数据分析编程平台。最后对 Python 的一些编程环境做了进一步介绍。

本书内容丰富，图文并茂，可操作性强且便于查阅，主要面向进行数据分析的读者，能有效地帮助读者提高数据处理与分析水平，提升工作效率。本书适合各层次的数据分析用户，既可作为初学者的入门指南，又可作为中高级用户的参考手册，同时也可作为各大中专院校和培训班的数据分析教材。

本书具有以下三大优点：

(1) 使用 Python 科学计算发行版 Anaconda，方便数据分析者使用。

读者可从 https://www.anaconda.com 下载安装并直接使用。

(2) 公开本书自定义函数的源代码，使用者可以深入理解 Python 函数的编程技巧，用这些函数建立自己的开发包；并建立了本书的学习博客（http://blog.leanote.com/DaPy），书中的例子数据、习题数据及相关代码都可直接在网上下载使用。

(3) 采用网络化教学平台。Python 的基础版缺少一个面向一般人群的菜单界面，这对那些只想用其进行数据分析的使用者而言是一大困难，本书采用流行的 Python 网络分析平台 Jupyter（https://jupyter.org），该平台可作为数据分析教学软件使用。

书中软件输出的坐标图多数没有标出横、纵坐标的量，目的是与软件界面保持一致。

本书在写作过程中得到了广东恒电信息科技股份有限公司的大力支持，该公司将为本书的实战操作提供可靠的实训环境支持，读者可以使用恒华大数据实训管理系统完成本书的实验操作。

本书由王斌会、王术共同完成，其中第 1～5 章由王斌会撰写，第 6～9 章由王术撰写，王斌会负责全书统稿。

由于作者知识和水平有限，书中难免有错误和不足之处，欢迎读者批评指正！

作　者

2019 年 1 月于暨南园

目　　录

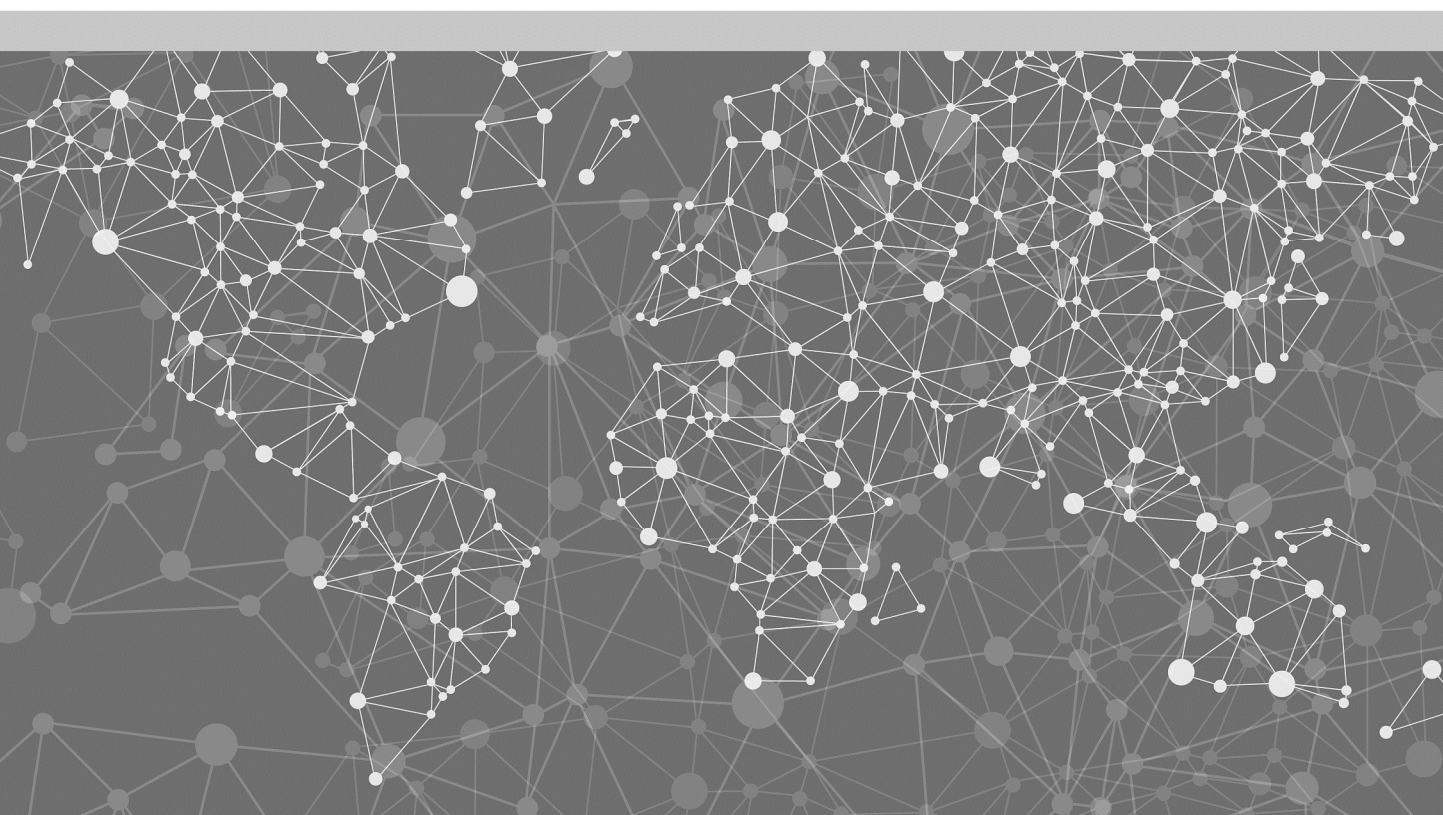
第一部分　数据分析基础知识

第二部分　数据分析高级方法

第三部分 大数据基本处理方法

第一部分

数据分析基础知识



第 1 章　数据收集与分析软件

1.1　数据收集过程

1.1.1　数据的类型

数据是采用某种计量尺度对事物进行计量的结果，采用不同的计量尺度会得到不同类型的数据。通常按数据的收集途径可将数据进行如下分类：

1.1.1.1　按度量尺度分

(1)定性数据(也称计数数据，qualitative data)

定性数据是对度量事物进行分类的结果。数据表现为类别，用文字来表述，如性别、区域、产品分类等。假如某班学生按性别分为男、女两类，那么性别就构成了一个定性变量。

性别：女，男，男，女，男，男，女，男，女，男，…，女，男，女，女，男，男，女，男，女

具体见 1.1.2 节例 1.1。

(2)定量数据(也称计量数据，quantitative data)

定量数据是对度量事物的精确测度。结果表现为具体的数值，如身高、体重、家庭收入、成绩等。假如测量某班每个学生的身高，这样身高就构成了一个定量变量。

身高：167，171，175，169，154，183，169，166，165，173，…，164，169，166，175，166，159，169，165

具体见 1.1.2 节例 1.1。

这类数据的详细分析参见王斌会编著的《数据统计分析及 R 语言编程》(第二版)。

1.1.1.2　按时间状况分

(1)横截面数据(也称截面数据，cross-section data)

横截面数据是指对变量在某一时点上收集的数据的集合，反映在相同或近似相同的时间点上收集的数据描述现象在某一时刻的变化情况。比如，2014 年我国各地区的国内生产总值、从业人员等数据：

地区	北京	天津	河北	山西	…	甘肃	青海	宁夏	新疆
生产总值	162.519	113.073	245.158	112.376	…	50.204	16.704	21.022	66.101
从业人员	1069.70	763.16	3962.42	1738.90	…	1500.30	309.18	339.60	953.34

当收集的数据有多个指标时，就形成了多元统计分析的数据格式，具体见 1.1.2 节例 1.2。

这类数据的详细分析参见王斌会编著的《多元统计分析及 R 语言建模》（第四版）。

(2)时间序列数据（也称动态数列，time series data）

时间序列数据是按照一定的时间间隔对某一变量在不同时间的取值进行观测得到的一组数据，反映在不同时间上收集到的数据描述现象随时间变化的情况。比如，收集 2015 年 6 月 3 日至 2018 年 5 月 31 日的沪深 300 指数的收盘价数据，这些数据就是一个时间序列数据：

日期 2015-6-3 2015-6-4 2015-6-5 2015-6-8 ... 2018-5-28 2018-5-29 2018-5-30 2018-5-31

收盘价 5143.590 5181.416 5230.552 5353.751 ... 3833.26 3804.01 3723.37 3802.38

具体见 1.1.2 节的例 1.3。

这类数据的详细分析参见王斌会编著的《计量经济学模型及 R 语言应用》一书。

1.1.2 数据的收集

数据收集有一定的格式，当对一个观察指标测量了每一观察单位的数据时，通常以向量的形式展现，x：$x_1, x_2, \cdots, x_n$。

当对每一观察单位测量了多个指标时，通常以双向表的矩阵形式展现，即

$$\boldsymbol{X}：\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_m$$

这里 $\boldsymbol{X}_j(j=1, 2, \cdots, m)$ 为 $n\times1$ 向量，$\boldsymbol{X}=(x_{ij})_{n\times m}$，如下所示。

$$\begin{array}{c} \\ 1 \\ 2 \\ \cdots \\ n \end{array}\begin{array}{c} \begin{array}{cccc} X_1 & X_2 & \cdots & X_m \end{array} \\ \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{11} & x_{22} & \cdots & x_{21} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \end{array}$$

不同领域对该数据的观察单位和指标的叫法不同：数学上称它们为行(row)和列(column)的二维数组或矩阵，统计学上称它们为观测(observation)和变量(variable)的数据集，数据库中称它们为记录(record)和字段(field)的数据表，人工智能中称它们为示例(example)和属性(attribute)的数据集。

为了使大家将注意力集中在如何进行数据分析，而不是将精力花在对数据的收集和输入上，本书采用一种新的数据分析策略，即通篇使用几组数据讲解如何进行数据分析。

1.1.2.1 单变量数据收集

这类数据通常都是一个个单独的数据变量，都可单独拿来进行数据分析。

【例 1.1　调查数据】

为了解某高校 52 名研究生的一些基本情况和对开设数据分析课程的一些看法，共收集了这些学生的八项指标(有时为了方便编程运算，也可将变量名改成英文或拼音形式)：

学生编号(定性变量，按年份、学院、专业、序号排列，简记为【学号】，也可记为 id)。

学生性别(定性变量，简记为【性别】，也可记为 sex)。

学生身高(定量变量，单位 cm，简记为【身高】，也可记为 height)。

学生体重(定量变量，单位 kg，简记为【体重】，也可记为 weight)。

学生个人年消费支出额(定量变量，单位千元，简记为【支出】，也可记为 outcome)。

开设课程的必要性(定性变量，简记为【开设】，也可记为 setup)。

是否学过相关课程(定性变量，简记为【课程】，也可记为 course)。

是否学过或用过何种数据分析软件(定性变量，简记为【软件】，也可记为 software)。

数据由变量及其观测值所组成。本例共有 8 个变量：学号、性别、身高、体重、支出、开设、课程、软件。

表 1-1 是 52 名研究生的个人和开课信息调查数据，按照该数据格式，每行为一个观测单位(样品)，每列为一个指标(变量)。于是就构成了表 1-1 的数据集，该数据保存在 PyDm_data.xlsx 文档的基本数据表单【BSdata】中。

表 1-1　52 名研究生的开课信息调查数据

学号	性别	身高	体重	支出	开设	课程	软件
1510248008	女	167	71	46.0	不清楚	都未学过	No
1510229019	男	171	68	10.4	有必要	概率统计	Matlab
1512108019	女	175	73	21.0	有必要	统计方法	SPSS
1512332010	男	169	74	4.9	有必要	编程技术	Excel
1512331015	男	154	55	25.9	有必要	都学习过	Python
1516248014	男	183	76	85.6	不必要	编程技术	Excel
1516352030	女	169	71	9.1	有必要	编程技术	Excel
1516171019	女	166	66	2.5	不必要	都未学过	Excel
1516391008	女	165	69	35.6	不必要	都未学过	Excel
1520395019	男	173	63	22.8	有必要	统计方法	R
1520100029	男	184	82	10.3	有必要	都学习过	SAS
1520324035	男	163	66	13.0	有必要	概率统计	Matlab
1522186005	男	162	63	9.8	有必要	都学习过	SPSS
1522160006	女	168	72	35.3	不必要	统计方法	SPSS
1522274026	女	164	66	50.5	有必要	统计方法	SPSS
1523376027	男	180	81	64.1	有必要	统计方法	Excel
1523368030	女	158	63	20.6	不清楚	都学习过	Excel
1524225006	男	179	75	5.8	有必要	编程技术	Python
1524105026	女	163	65	69.4	有必要	编程技术	Python
1524286013	男	160	62	4.8	有必要	都未学过	R

续表

学号	性别	身高	体重	支出	开设	课程	软件
1525235027	女	168	70	8.2	有必要	都学习过	R
1525352033	男	185	83	5.1	有必要	都学习过	SPSS
1526177005	男	174	76	15.8	有必要	概率统计	Excel
1526196010	男	167	72	9.8	不清楚	统计方法	SPSS
1527173011	女	160	62	11.5	不必要	都学习过	Matlab
1527237032	女	163	65	19.4	有必要	统计方法	R
1527289024	男	155	50	10.8	有必要	概率统计	SPSS
1529107020	男	178	78	8.9	不清楚	概率统计	Matlab
1529314037	男	170	70	15.1	有必要	概率统计	SAS
1529245023	男	164	58	21.9	有必要	统计方法	Excel
1529365032	男	172	71	10.4	有必要	都学习过	SPSS
1530273031	男	178	77	35.6	不必要	统计方法	R
1530243029	男	186	87	9.5	不必要	都未学过	No
1531364037	女	171	69	7.3	有必要	都学习过	Excel
1531316038	女	156	56	52.8	有必要	统计方法	Excel
1532304031	女	166	68	47.9	不清楚	统计方法	SAS
1532208040	男	176	78	75.5	不必要	概率统计	Excel
1532292012	男	178	78	28.4	不必要	概率统计	No
1532185004	女	155	54	13.4	不清楚	编程技术	Excel
1533219013	女	163	62	11.1	不清楚	概率统计	Matlab
1533384028	男	158	60	6.1	有必要	编程技术	R
1533172017	女	167	68	27.2	不必要	都未学过	Excel
1537288004	女	173	70	19.1	不清楚	编程技术	Python
1537359035	女	174	71	17.6	不清楚	概率统计	No
1438391022	女	164	62	10.3	有必要	编程技术	Python
1538399025	男	169	65	9.5	有必要	统计方法	SAS
1438120022	男	166	70	35.6	有必要	统计方法	R
1538319004	男	175	68	44.4	不清楚	统计方法	SAS
1538254010	女	166	65	5.3	不清楚	编程技术	Python
1540294017	女	159	58	71.4	不清楚	都学习过	SPSS
1540365026	女	169	73	5.5	有必要	统计方法	Excel
1540388036	女	165	67	56.8	不必要	概率统计	SAS

1.1.2.2 多元数据收集

这类数据也称横截面数据，主要用来研究多个变量间的关系，包括综合分析、分类分析等。

【例 1.2 综合数据】

为了解我国各地区对外贸易国际竞争力的情况，我们从各省(市、自治区)的对外贸易能力、对外贸易经济效益、贸易资本竞争力等方面选取了 8 个对外贸易国际竞争力的基础指标。

- 地区国内生产总值(百亿元，简记为【生产总值】，也可记为 Y)
- 从业人员人数(万人，简记为【从业人员】，也可记为 X1)
- 全社会固定资产投资额(百亿元，简记为【固定资产】，也可记为 X2)
- 实际利用外资总额(百亿元，简记为【利用外资】，也可记为 X3)
- 进出口贸易总额(亿美元，简记为【进出口额】，也可记为 X4)
- 工业企业新产品出口额(亿元，简记为【新品出口】，也可记为 X5)
- 国际市场占有率(‰，简记为【市场占有】，也可记为 X6)
- 对外贸易依存度(%，简记为【对外依存】，也可记为 X7)

这些指标基本覆盖了各省外贸国际竞争力的各方面，能够较好地反映各省国际竞争力水平。具体数据如表 1-2 所示。

表 1-2　我国 30 个省、市、自治区 2011 年对外贸易数据

地区	生产总值	从业人员	固定资产	利用外资	进出口额	新品出口	市场占有	对外依存
北京	162.519	1069.70	55.789	196.906	3894.9	6470.51	2.635	1.55
天津	113.073	763.16	70.677	61.947	1033.9	7490.32	1.986	0.59
河北	245.158	3962.42	163.893	178.782	536.0	2288.19	1.276	0.14
山西	112.376	1738.90	70.731	104.945	147.6	1522.79	0.242	0.08
内蒙古	143.599	1249.30	103.652	54.426	119.4	342.36	0.209	0.05
辽宁	222.267	2364.90	177.263	155.296	959.6	4150.24	2.278	0.28
吉林	105.688	1337.80	74.417	58.843	220.5	746.94	0.223	0.13
黑龙江	125.820	1977.80	74.754	81.979	385.1	318.89	0.789	0.20
上海	191.957	1104.33	49.621	179.582	4373.1	10326.44	9.359	1.47
江苏	491.103	4758.23	266.926	261.118	5397.6	43928.94	13.953	0.71
浙江	323.189	3680.00	141.853	239.452	3094.0	25355.08	9.657	0.62
安徽	153.007	4120.90	124.557	92.613	313.4	2344.05	0.762	0.13
福建	175.602	2459.99	99.109	92.158	1435.6	7957.50	4.144	0.53
江西	117.028	2532.60	90.876	71.531	315.6	1301.04	0.977	0.17
山东	453.619	6485.60	267.497	223.057	2359.9	17688.02	5.614	0.34
河南	269.310	6198.00	177.690	147.022	326.4	2176.17	0.859	0.08
湖北	196.323	3672.00	125.573	113.434	335.2	1614.37	0.872	0.11
湖南	196.696	4005.03	118.809	106.234	190.0	1814.50	0.442	0.06
广东	532.103	5960.74	170.692	410.616	9134.8	56849.07	23.742	1.11
广西	117.209	2936.00	79.907	66.822	233.5	641.55	0.556	0.13
海南	25.227	459.22	16.572	18.885	127.6	185.49	0.113	0.33
重庆	100.114	1590.16	74.734	70.117	292.2	3928.45	0.886	0.19
四川	210.267	4785.50	142.222	162.007	477.8	1233.51	1.297	0.15
贵州	57.018	1792.80	42.359	39.441	48.8	308.65	0.134	0.06

续表

地区	生产总值	从业人员	固定资产	利用外资	进出口额	新品出口	市场占有	对外依存
云南	88.931	2857.24	61.910	66.849	160.5	257.76	0.423	0.12
陕西	125.123	2059.02	94.311	92.209	146.2	408.45	0.313	0.08
甘肃	50.204	1500.30	39.658	42.500	87.4	300.89	0.096	0.11
青海	16.704	309.18	14.356	10.488	9.2	0.30	0.030	0.04
宁夏	21.022	339.60	16.447	13.563	22.9	197.00	0.071	0.07
新疆	66.101	953.34	46.321	44.409	228.2	83.39	0.751	0.22

本书所选数据是中国 30 个省(市、自治区)(未包括西藏)2011 年的相关数据，数据来源于中国统计年鉴和各省统计年鉴，该数据存放在 PyDm_data.xlsx 文档的多元数据【MVdata】表单中。

1.1.2.3 时序数据的收集

时序数据是一类比较特殊的数据，也称为纵向数据，它对数据的格式有一定要求，特别是时间序列数据，须注意时间序列数据的输入格式。

【例 1.3 日期数据—股票数据】

今从某证券网站收集到 2015 年 6 月 3 日至 2018 年 5 月 31 日三年的沪深 300 指数的收盘价数据，如表 1-3 所示。这是一种典型的日期时间序列数据集，共 3 年 732 个数据，该数据存放在 PyDm_data.xlsx 文档的股票数据【TSdata】表中。

表 1-3 沪深 300 日收盘价数据

日 期	收 盘 价	日 期	收 盘 价	日 期	收 盘 价
2015-6-3	5143.590	2017-5-2	3426.58	…	…
2015-6-4	5181.416	2017-5-3	3413.13	2018-5-18	3903.06
2015-6-5	5230.552	2017-5-4	3404.39	2018-5-21	3921.24
2015-6-8	5353.751	2017-5-5	3382.55	2018-5-22	3906.21
2015-6-9	5317.461	2017-5-8	3358.81	2018-5-23	3854.58
2015-6-10	5309.112	2017-5-9	3352.53	2018-5-24	3827.22
2015-6-11	5306.590	2017-5-10	3337.70	2018-5-25	3816.50
2015-6-12	5335.115	2017-5-11	3356.65	2018-5-28	3833.26
2015-6-15	5221.167	2017-5-12	3385.38	2018-5-29	3804.01
2015-6-16	5064.820	2017-5-15	3399.19	2018-5-30	3723.37
…	…	2017-5-16	3428.65	2018-5-31	3802.38

进一步，我们还可以收集股票指数的时数据、分数据、秒数据、毫秒数据和微秒数据，这类数据就形成了高频数据，是一种大数据，限于篇幅，本文将不涉及。

上述的数据都是一些结构化数据，但随着大数据时代的来临，出现了大量的非结构化数据，这些数据的类型不只是由数字构成的数据库，还包括大量的文字、图像、影像和视频数据。

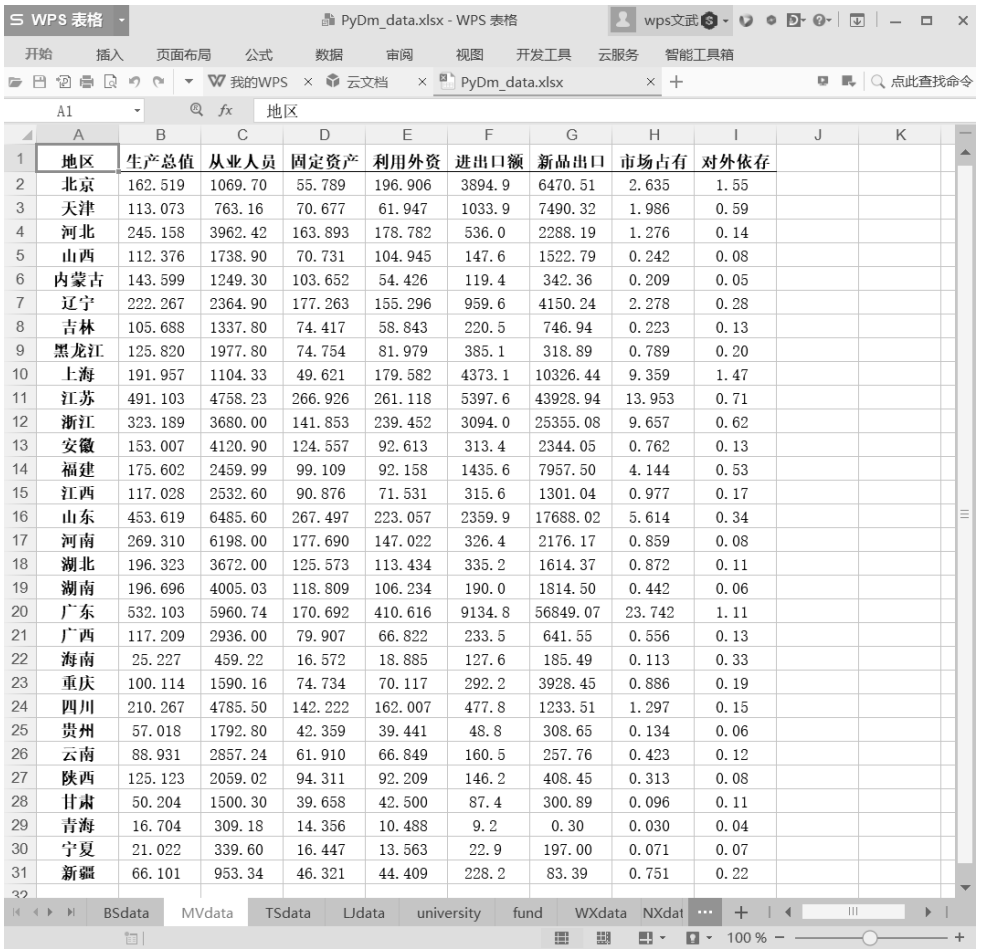
1.1.3 数据的管理

数据管理是利用计算机硬件和软件技术对数据进行有效的收集、存储、处理和应用的过程。对于一般的数据分析而言，电子表格软件已经足以胜任分析所需要的数据管理。最常用的电子表格软件有微软 Office 的 Excel 表格软件(收费)和金山 Office 的 WPS 表格软件(免费)。

1.1.3.1 电子表格管理数据

如果仅做一般数据管理，数据量不是特别大，而且要求系统免费、跨平台，那么首选的数据管理软件应该是 WPS 表格软件(WPS 表格是跟 Excel 兼容度最高的电子表格软件，但 WPS 是免费的，建议使用)。下面是采用 WPS 表格对上面数据的管理界面。

数据存放在 PyDm_data.xlsx 文档中，可登录 blog.leanote.com/PyDm 下载该数据。



地区	生产总值	从业人员	固定资产	利用外资	进出口额	新品出口	市场占有	对外依存
北京	162.519	1069.70	55.789	196.906	3894.9	6470.51	2.635	1.55
天津	113.073	763.16	70.677	61.947	1033.9	7490.32	1.986	0.59
河北	245.158	3962.42	163.893	178.782	536.0	2288.19	1.276	0.14
山西	112.376	1738.90	70.731	104.945	147.6	1522.79	0.242	0.08
内蒙古	143.599	1249.30	103.652	54.426	119.4	342.36	0.209	0.05
辽宁	222.267	2364.90	177.263	155.296	959.6	4150.24	2.278	0.28
吉林	105.688	1337.80	74.417	58.843	220.5	746.94	0.223	0.13
黑龙江	125.820	1977.80	74.754	81.979	385.1	318.89	0.789	0.20
上海	191.957	1104.33	49.621	179.582	4373.1	10326.44	9.359	1.47
江苏	491.103	4758.23	266.926	261.118	5397.6	43928.94	13.953	0.71
浙江	323.189	3680.00	141.853	239.452	3094.0	25355.08	9.657	0.62
安徽	153.007	4120.90	124.557	92.613	313.4	2344.05	0.762	0.13
福建	175.602	2459.99	99.109	92.158	1435.6	7957.50	4.144	0.53
江西	117.028	2532.60	90.876	71.531	315.6	1301.04	0.977	0.17
山东	453.619	6485.60	267.497	223.057	2359.9	17688.02	5.614	0.34
河南	269.310	6198.00	177.690	147.022	326.4	2176.17	0.859	0.08
湖北	196.323	3672.00	125.573	113.434	335.2	1614.37	0.872	0.11
湖南	196.696	4005.03	118.809	106.234	190.0	1814.50	0.442	0.06
广东	532.103	5960.74	170.692	410.616	9134.8	56849.07	23.742	1.11
广西	117.209	2936.00	79.907	66.822	233.5	641.55	0.556	0.13
海南	25.227	459.22	16.572	18.885	127.6	185.49	0.113	0.33
重庆	100.114	1590.16	74.734	70.117	292.2	3928.45	0.886	0.19
四川	210.267	4785.50	142.222	162.007	477.8	1233.51	1.297	0.15
贵州	57.018	1792.80	42.359	39.441	48.8	308.65	0.134	0.06
云南	88.931	2857.24	61.910	66.849	160.5	257.76	0.423	0.12
陕西	125.123	2059.02	94.311	92.209	146.2	408.45	0.313	0.08
甘肃	50.204	1500.30	39.658	42.500	87.4	300.89	0.096	0.11
青海	16.704	309.18	14.356	10.488	9.2	0.30	0.030	0.04
宁夏	21.022	339.60	16.447	13.563	22.9	197.00	0.071	0.07
新疆	66.101	953.34	46.321	44.409	228.2	83.39	0.751	0.22

1.1.3.2 数据库管理数据

当分析的数据量很大时，采用电子表格类软件有很大问题，须采用数据库来管理数据表格。

1.2 数据分析软件

1.2.1 数据分析软件简介

能做数据分析的软件有很多，如电子表格、SAS、SPSS、R、Python、Stata、Matlab、Eviews 等，下面简单介绍一下这些软件。

电子表格(Excel、WPS 等)不仅是数据管理软件，也是分析数据的入门工具。尽管其统计分析功能并不十分强大，但是它可以快速地做一些基本的数据分析工作，也可创建供大多数人使用的数据图表。由于电子表格在数据存量、图形样式、统计方法和统计建模方面功能受限，所以它们很难成为专业的数据分析软件。

SAS(Statistics Analysis System)是使用最为广泛的三大著名统计分析软件(SAS，SPSS 和 Splus)之一，被誉为统计分析的标准软件。SAS 是功能最为强大的统计软件，有完善的数据管理和统计分析功能，是熟悉统计学并擅长编程的专业人士的首选。

SPSS(Statistical Package for the Social Science)也是世界上著名的统计分析软件之一。SPSS 中文名为社会科学统计软件包，这是为了强调其社会科学应用的一面，而实际上它在社会科学和自然科学的各个领域都能发挥巨大作用。与 SAS 比较，SPSS 是非统计学专业人士的首选。

Matlab 是美国 MathWorks 公司出品的商业数学软件，是用于算法开发、数据可视化、数据分析及数值计算的高级技术计算语言和交互式环境，主要包括 Matlab 和 Simulink 两大部分。它在数值计算和模拟分析方面首屈一指，主要应用于工程计算、控制设计、信号处理与通信、图像处理、信号检测、金融建模设计与分析等领域。

Stata 是一套完整的、集成的统计分析软件包，可以满足数据分析、数据管理和统计图形的所有需要。Stata 12 增加了许多新的特征，比如结构方程模型(SEM)、ARFIMA、Contrasts、ROC 分析、自动内存管理等。Stata 适用于 Windows、Macintosh 和 Unix 平台计算机(包括 Linux)。Stata 的数据集、程序和其他的数据能够跨平台共享，且不需要转换，同样可以快速而方便地从其他统计软件包、电子表单和数据库中导入数据集。

Eviews 是美国 QMS 公司 1981 年发行的第 1 版 Micro TSP 的 Windows 版本，通常称为计量经济学软件包，是当今世界最流行的计量经济学软件之一。它可应用于科学计算中的数据分析与评估、财务分析、宏观经济分析与预测、模拟、销售预测和成本分析等。由于 Eviews 提供了一个很好的工作环境，能够迅速地进行编程、估计、使用新的工具和技术，所以它在计量经济建模方面有着广泛的应用。

从纯数据分析角度来说，应用最好的当属 S 语言的免费开源及跨平台系统 R 语言。R 语言是一个用于统计计算的很成熟的免费软件，也可以把它理解为一种统计计算语言，实际上很多人都直接称呼它为“R”，它比 C++，Fortran 等不知道简单了多少倍！如果你是一位数据分析的初学者，面对众多数据分析软件感到困惑且难以抉择，又想快速地掌

握统计计算、数据分析甚至目前比较流行的数据挖掘技术，那么首选的语言就是 R。

不过，R 语言对于初学编程和数据分析的人来说，入门还是有一定难度的，因为它还不是真正意义上的一般编程语言，所以现在流行“人生苦短，我用 Python”这样的说法，说明 Python 作为一种新兴的编程语言，已深入人心。现在我国许多地区高考试卷中都加入了 Python 编程的内容，一些中小学也开始开设 Python 编程课程。另外，由于 Python 博采众长，不断吸收其他数据分析软件的优点，并加入了大量的数据分析功能，它已成为仅次于 Java、C 及 C++的第四大语言，且在数据处理领域有超过 R 语言的趋势，所以本数据分析教程采用了 Python 作为分析工具。

综上所述，出于数据管理的方便，适用于一般数据分析的最好的数据管理软件应该是电子表格类软件(如微软 Office 的 Excel，金山 WPS 的表格等)，大量数据可以在一个工作簿中保存。所以，对于规模不是非常大的数据集，建议采用该方法来管理和编辑数据，而统计软件是我们进行数据分析不可或缺的工具。随着知识产权保护要求的不断提高，免费和开放源代码逐渐成为一种趋势，Python 正是在这个大背景下发展起来的，并逐渐成为数据分析的标准软件。考虑到微软的 Excel 必须购买正版，而 WPS 表格提供官方免费正版软件，笔者认为，通常的数据处理和分析工作用 WPS+Python 足矣。

1.2.2 Python 语言介绍

1.2.2.1 Python 简介

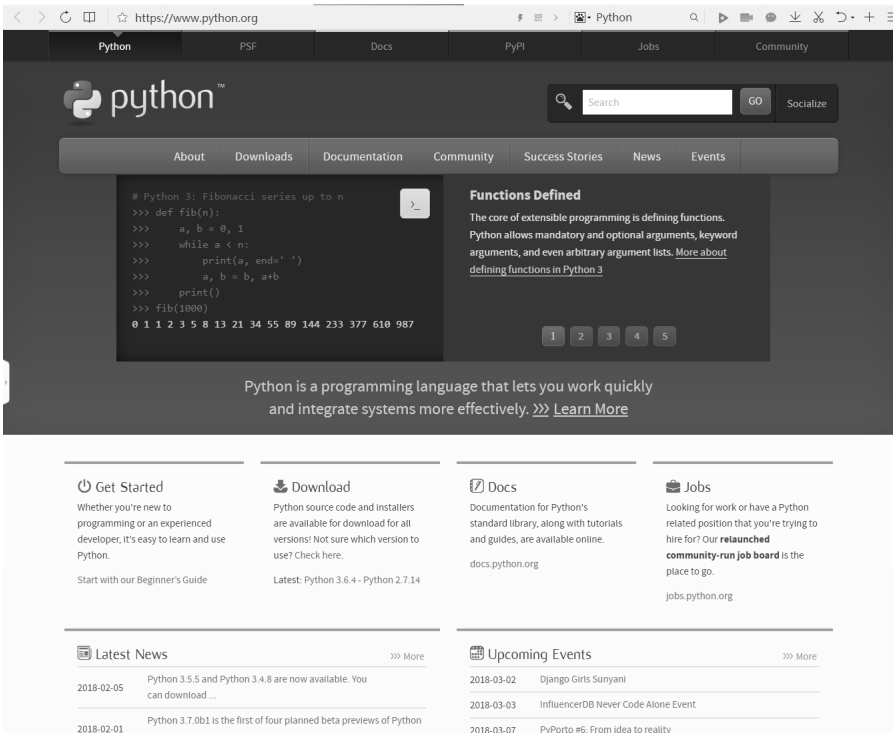
Python(英国发音：/'paɪθən/，美国发音：/'paɪθɑːn/)，是一种面向对象的解释型计算机程序设计语言，由荷兰人 Guido van Rossum 于 1989 年发明，第一个公开发行版发行于 1991 年。

Python 是纯粹的自由软件，源代码和解释器 CPython 遵循 GPL(GNU General Public License)协议。Python 语法简洁清晰，特色之一是强制用空白符(white space)作为语句缩进。

Python 具有丰富而强大的包，它常被昵称为“胶水语言”，能够把用其他语言制作的各种模块(尤其是 C/C++)轻松地联结在一起。常见的一种应用情形是，使用 Python 快速生成程序的原型(有时甚至是程序的最终界面)，然后对其中有特别要求的部分，用更合适的语言改写，比如，3D 游戏中的图形渲染模块性能要求特别高，就可以用 C/C++重写，然后封装为 Python 可以调用的扩展包。需要注意的是，在使用扩展包时可能需要考虑平台问题，某些扩展包可能不提供跨平台的实现。

由于 Python 语言的简洁性、易读性及可扩展性，在国外用 Python 做科学计算的研究机构日益增多，一些知名大学已经采用 Python 来教授程序设计课程。例如，卡耐基梅隆大学的编程基础、麻省理工学院的计算机科学及编程导论就使用 Python 语言讲授。众多开源的科学计算软件包都提供了 Python 的调用接口，如著名的计算机视觉包 OpenCV、三维可视化包 VTK、医学图像处理包 ITK。而 Python 专用的科学计算扩展包就更多了，如以下三个十分经典的科学计算扩展包：numpy、scipy 和 Matplotlib，它们分别为 Python 提供了快速数组处理、数值运算及绘图功能。因此，Python 语言及其众多的扩展包所构

成的开发环境十分适合工程技术、科研人员处理实验数据、制作图表，甚至开发科学计算应用程序。Python 的官方网站为 https://www.python.org/，在该网站可以下载 Python 软件和许多程序包，以及有关 Python 的资料。



1.2.2.2 Python 的特色

Python 是一种高层次的结合了解释性、编译性、互动性和面向对象的脚本语言，其设计具有很强的可读性。

① Python 是解释型语言：这意味着开发过程中没有了编译这个环节。

② Python 是交互式语言：这意味着可以在一个 Python 提示符下直接互动执行写程序。

③ Python 是面向对象语言：这意味着 Python 支持面向对象的风格或代码封装在对象中的编程技术。

④ Python 是初学者的语言：Python 对初级程序员而言，是一种友好易学的语言，它支持广泛的应用程序开发——从简单的文字处理到 WWW 浏览器再到游戏。

具体而言，Python 有如下一些特点。

① 简单、易学。

② 免费、开源。

③ 高层语言：封装内存管理等。

④ 可移植性：程序如果不使用依赖于系统的特性，那么无须修改就可以在任何平台上运行。

⑤ 解释性：直接从源代码运行程序，不再需要担心如何编译程序，使得程序更加易于移植。

⑥ 面向对象：支持面向过程的编程，也支持面向对象的编程。

⑦ 可扩展性：需要保密或者高效的代码，可以用 C 或 C++编写，然后在 Python 程序中使用。

⑧ 可嵌入性：可以把 Python 嵌入 C/C++程序，从而向程序用户提供脚本功能。

⑨ 丰富的包：包括正则表达式、文档生成、单元测试、线程、数据库、网页浏览器、CGI、FTP、电子邮件、XML、XML-RPC、HTML、WAV 文件、密码系统、GUI(图形用户界面)、Tk 和其他与系统有关的操作。

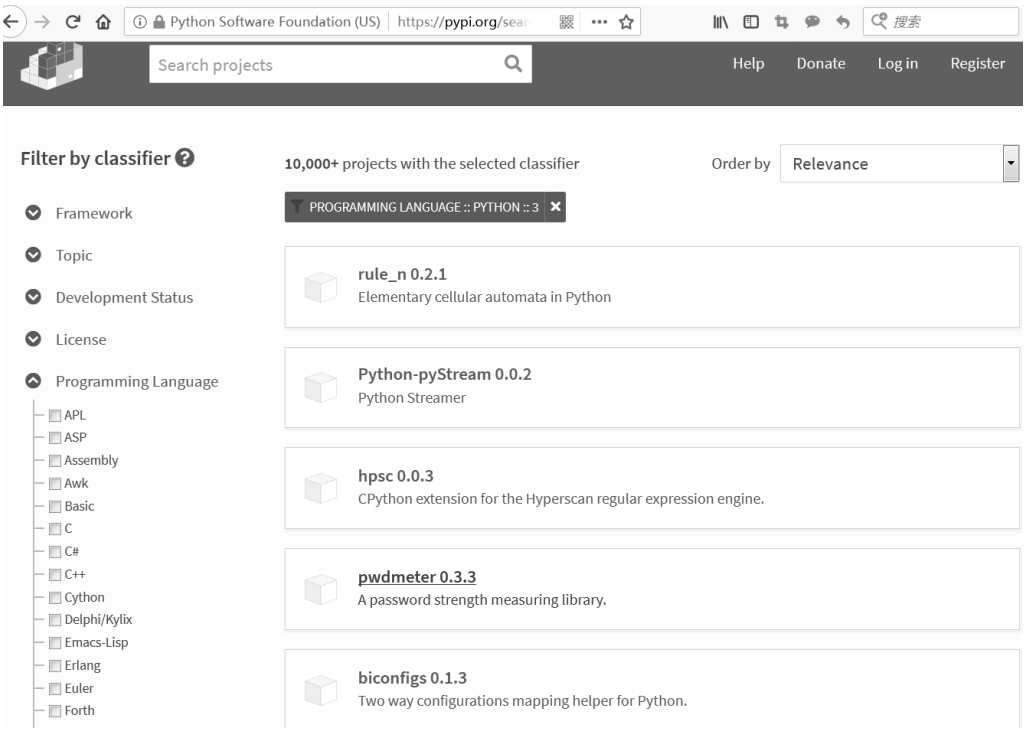
除标准包以外，还有许多其他高质量的包，如 wxPython、Twisted 和 Python 图像包等。

⑩ 概括性强：Python 确实是一种十分精彩又强大的语言，它合理地结合了高性能与使得编写程序简单有趣的特色。

⑪ 规范的代码：Python 采用强制缩进的方式，使得代码具有极佳的可读性。

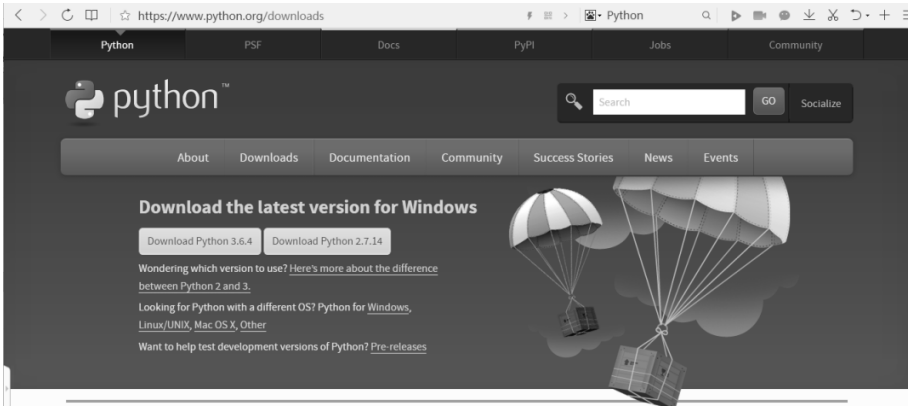
1.2.2.3 Python 的功能

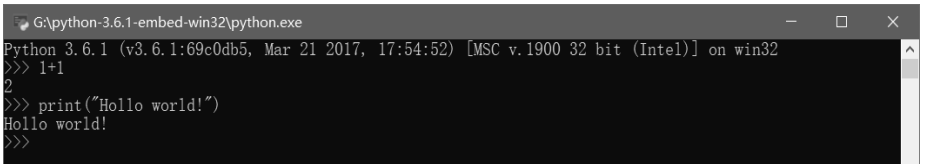
Python 最大也是其成为最流行的数据分析软件的特点就是，它包含大量的扩展包并拥有方便的二次开发功能。Python 的扩展包包罗万象，它所能完成的数据统计模型已经超出了任何其他商业统计软件。笔者做了一个统计，截至 2019 年 1 月，https://www.python.org/所列的扩展包达到 165797 个之多(包含几十万个数据分析方法)，除进行各种程序开发外，可完全满足进行数据分析之用。



1.2.2.4 Python 编程环境

Python 是一种强大的面向对象的编程语言，这样的编程环境需要使用者不仅熟悉各种命令的操作，还须熟悉 DOS 编程环境，而且所有命令执行完即进入新的界面，这给那些不具备编程经验或对统计方法掌握不够好的使用者造成了极大的困难。从 https://www.python.org/ 下载 Python 最新版，安装后只是一个不包括大量包的最基本的语言环境。本书采用基于 Anaconda 的 Jupyter 平台进行数据分析。

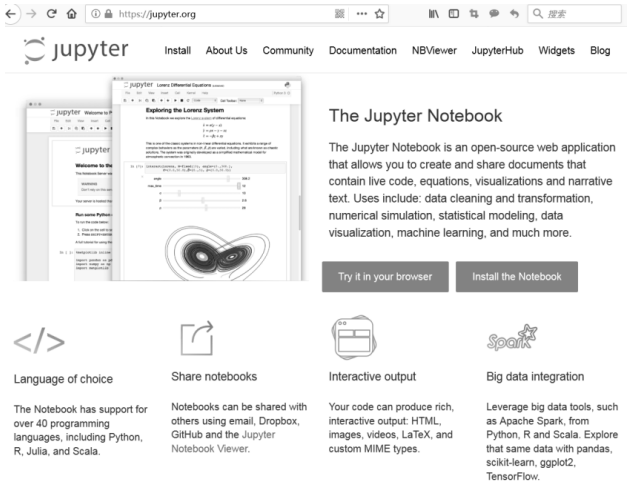




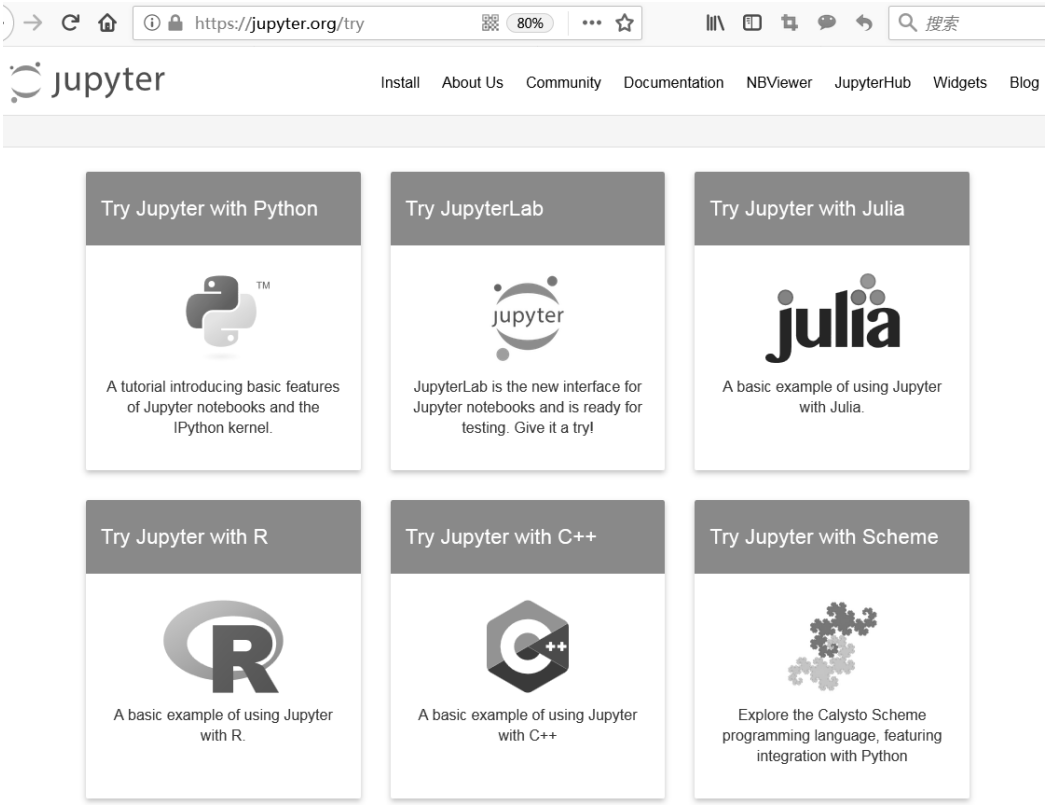
1.2.3 Python 在线平台

1.2.3.1 Jupyter 项目

随着网络技术的不断普及，建立基于大数据和云计算的 Web 应用平台势在必行。Jupyter 项目旨在开发跨几十种编程语言的开源软件、开放标准和用于交互式计算的服务。



Jupyter 项目目前提供了一个在线使用开源计算程序的云服务平台，可帮助大家快速使用 40 种以上编程语言，包括 Python、R、Julia 和 Scala 等，只要在网址中输入 https://jupyter.org/try 即可。



1.2.3.2 Jupyter Notebook

(1) Jupyter Notebook 简介

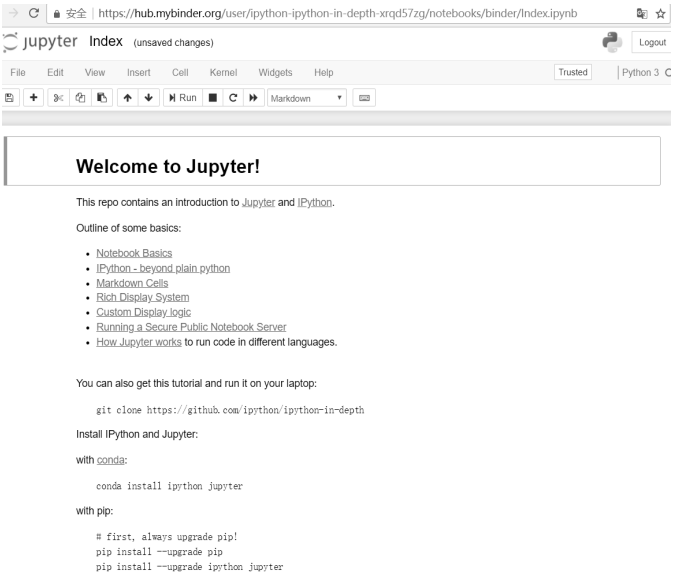
Jupyter Notebook 是一款开放源代码的 Web 应用程序，允许创建和共享包含实时代码、方程式、可视化和叙述文本的文档。用途包括数据清理和转换、数值模拟、统计建模、数据可视化、机器学习等。

使用 Jupyter Notebook，用户可以通过电子邮件、Dropbox、GitHub 和 Jupyter Notebook Viewer，将 Jupyter Notebook 分享给其他人。在 Jupyter Notebook 中，代码可以实时生成图像、视频、LaTeX 和 JavaScript。

数据挖掘领域的热门比赛 Kaggle 里的资料都是 Jupyter 格式的，本书也采用 Jupyter Notebook 格式。

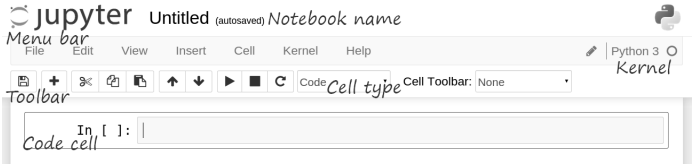
(2) Jupyter Notebook 的使用

Jupyter 社区提供浏览器版的 Jupyter Notebook，使用非常直观和方便，强烈推荐练习使用！但浏览器版只包含常用的程序包，一些复杂的程序包还得在本地安装版中使用。

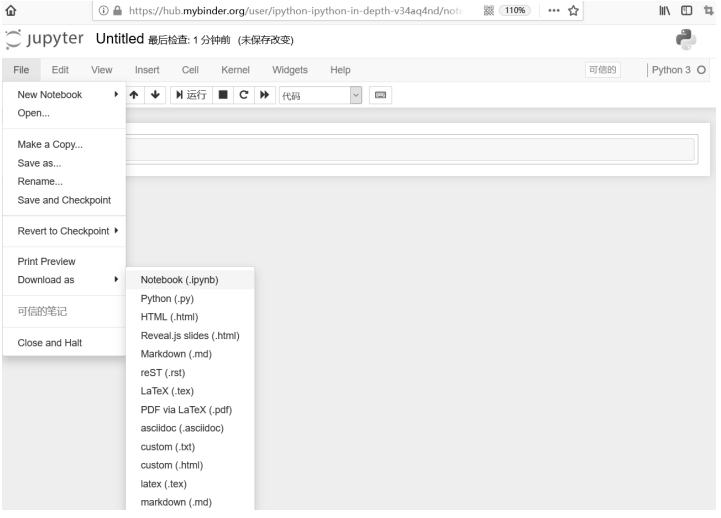


(3)新建 Jupyter Notebook 文档

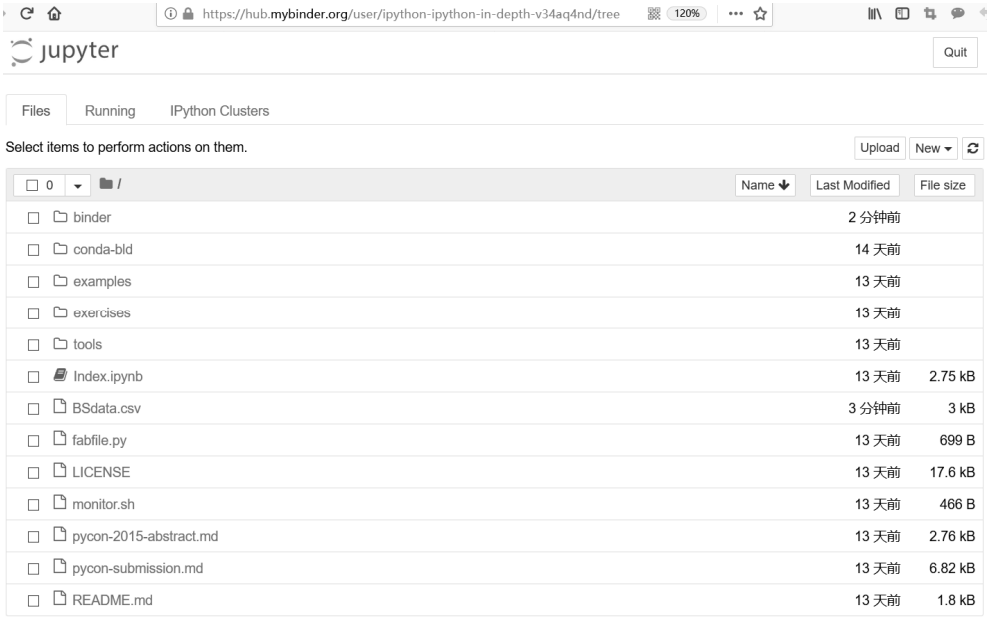
单击【New】按钮可建立相应的 Jupyter Notebook 文档语言文本，本书使用的是 Python3。建好文档后(这里默认文挡名为 Untitled.ipynb)就可以用 Python3 进行计算和分析了，也可以先建目录(Folder)，再建文档。



写文档时，cell 类型分成 markdown 和 code，可任意切换，直接写出；进行科学运算和画图时，numpy，scipy，pandas 等包以前都需要安装，现在全不用安装了。

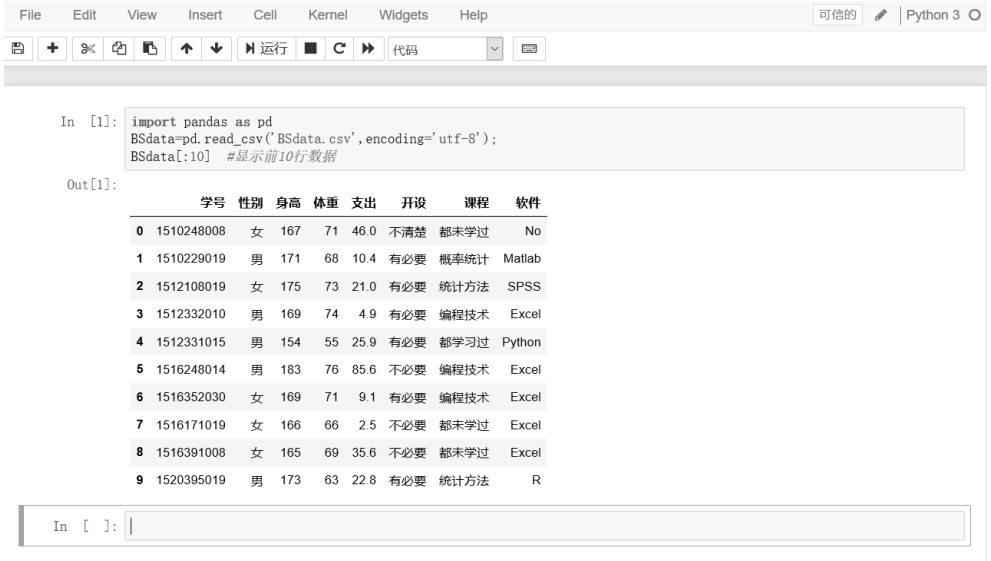


可以在文件管理菜单中修改(Rename...)之前新建的文档名，如将 Untitled.ipynb 修改为 myPython1.ipynb。也可以在文档菜单中下载并保存该文档，以备后用。



(4)上传文档与数据

由于 jupyter.org/try 是一个网络浏览器版，所以要使用自己的文档或数据，须事先上传【Upload】。比如，要用书中的基本数据进行分析，须上传 BSdata.csv 数据文档，然后就可以在 Jupyter Notebook 中使用了！



注意：对于文本数据，要留心数据的编码(encoding)格式！如果有中文名，要用'gb2312'或'utf-8'，但都得事先定义好！

(5) Jupyter Notebook 快捷键

Jupyter Notebook 有两种键盘输入模式。

① 编辑模式，允许往单元中输入代码或文本；这时的单元框线是绿色的。

② 命令模式，通过键盘输入运行程序命令；这时的单元框线是灰色的。

Shift+Enter：运行本单元，选中下一个单元；

Ctrl+Enter：运行本单元；

Alt+Enter：运行本单元，在其下插入新单元；

Y：单元转入代码状态；

M：单元转入 markdown 状态；

A：在上方插入新单元；

B：在下方插入新单元；

X：剪切选中的单元；

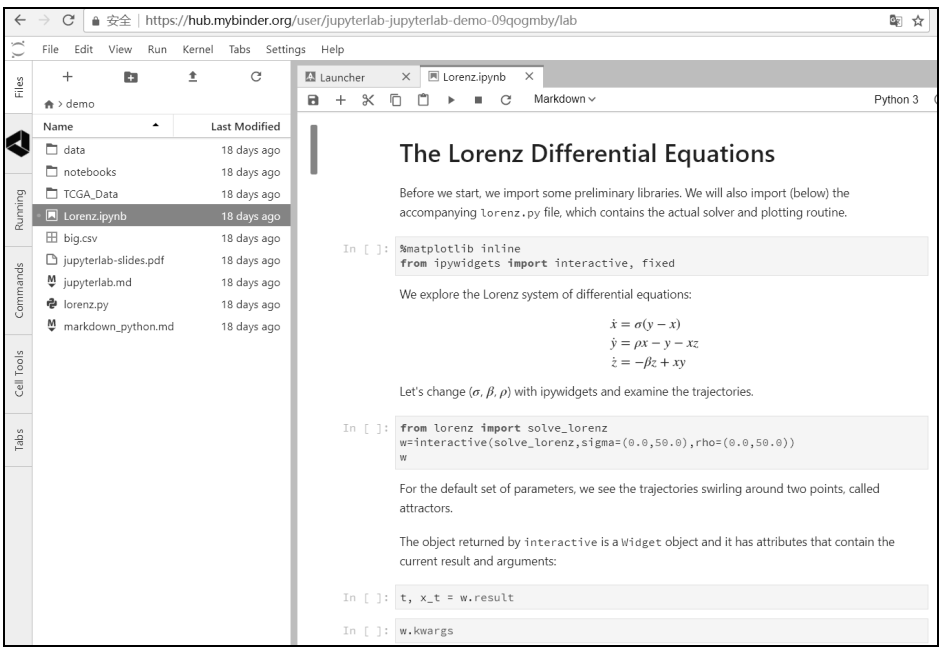
Shift +V：在上方粘贴单元。

这些快捷键也可在下面的 Jupyter Lab 中使用。

1.2.3.3 Jupyter Lab

Jupyter Lab 是一个名副其实的 IDE，且是一个基于网页的 IDE(保留了全部的 Notebook 特性)。

如果不想安装庞大的 Python 和 Jupyter Notebook，而只是想简单使用一下，那么可用 Jupyter 社区提供的浏览器版 Jupyter Lab，单击“试试 Jupyter Lab”即可使用，但限于网速，在线运行速度稍慢，建议使用本地安装版。



进入后的界面与平常使用的编程环境差别不大。

1.3 Python 编程基础

网上有大量的 Python 编程基础知识介绍，如

```
http://www.runoob.com/Python/Python-dictionary.html
```

请大家自行学习。由于本书重点介绍 Python 的数据分析，所以对 Python 编程的基础知识将不展开讨论。

1.3.1 Python 编程入门

1.3.1.1 Python 的工作目录

在使用 Python 时，一个重要设置是定义工作目录，即设置当前运行路径(全部数据和程序都将在该目录下工作)。例如，可以将 Python 工作目录设定为 D:\PyDm(先在 D 盘上建立目录 PyDm，然后在编程环境中使用)。

In [1]	#获得当前目录 %pwd #改变工作目录 %cd "D:\\PyDm" %pwd
Out [1]	"C:\user\1" D:\PyDm "D:\\PyDm"

1.3.1.2 Python 分析包

Python 具有丰富的数据分析模块，大多数做数据分析的人使用 Python 是因为其强大的数据分析功能。所有的 Python 函数和数据集是保存在包里面的。只有当一个包被安装并被载入(import)时，它的内容才可以被访问。这样做，一是为了高效(完整的列表会耗费大量的内存并且增加搜索的时间)；二是为了帮助包的开发者，防止命名和其他代码中的名称冲突。

由于 Anaconda 发行版已安装常用的数据分析包，所以我们只须调用即可。下面介绍几个 Python 常用的数据分析包，如表 1-4 所示。

表 1-4 Python 常用数据分析包

包 名	说 明	主要功能
math	基础数学包	提供函数，完成各种数学运算
random	随机数生成包	Python 中的 random 模块用于生成各种随机数
numpy	数值计算包	numpy(numeric python)是 Python 的一种开源的数值计算扩展，一个用 Python 实现的数值计算工具包。它提供许多高级的数值编程工具，如矩阵数据类型、矢量处理，以及精密的运算包。专为进行严格的数值处理而产生

续表

包　名	说　明	主要功能
scipy	数值分析包	提供很多科学计算工具包和算法，方便是易于使用，专为科学和工程设计的数值分析工具包。它包括统计、优化、整合、线性代数模块、傅里叶变换、信号和图像处理、常微分方程求解器等，包含常用的统计估计和检验方法
pandas	数据操作包	提供类似于 R 语言的 DataFrame 操作，非常方便。pandas 是面板数据(panel data)的简写。它是 Python 最强大的数据分析和探索工具，因金融数据分析工具而开发，支持类似 SQL 的数据增、删、改、查，支持时间序列分析，灵活处理缺失数据
statsmodels	统计模型包	statsmodels 可以补充 scipy.stats，是一个包含统计模型、统计测试和统计数据挖掘的 Python 模块。对每个模型都会生成一个对应的统计结果，对时间序列有完美的支持
matplotlib	基本绘图包	该包主要用于绘图和绘表，是一个强大的数据可视化工具，语法类似于 Matlab，是一个 Python 的图形框架，类似于 Matlab 和 R 语言。它是 Python 最著名的绘图库，提供了一整套和 Matlab 相似的命令 API，十分适合交互式制图。而且也可以方便地将它作为绘图控件，嵌入 GUI 应用程序中
sklearn	机器学习包	sklearn 是基于 Python 的机器学习工具模块，里面主要包含 6 大模块：分类、回归、聚类、降维、模型选择、预处理，如，使用 sklearn.decomposition 可进行主成分分解
beautifulSoup	网络爬虫包	beautifulsoup 是 Python 的一个包，最主要的功能是从网页抓取数据。BeautifulSoup 提供一些简单的、Python 式的函数，用来处理导航、搜索、修改分析树等功能。通过解析文档为用户提供需要抓取的数据，通过它可以很方便地提取出 HTML 或 XML 标签中的内容
networkx	复杂网络包	networkx 是一款 Python 的软件包，用于创造、操作复杂网络，以及学习复杂网络的结构、动力学及其功能。 通过它可以用标准或者不标准的数据格式加载或者存储网络，它可以产生许多种类的随机网络或经典网络，也可以分析网络结构、建立网络模型、设计新的网络算法、绘制网络等

注意：安装程序包和载入程序包是两个概念，安装程序包是指将需要的程序包安装到电脑中，载入包是指将程序包调入 Python 环境中。程序包的安装(通常在命令行状态：)>>>pip install pandas。

Python 调用包的命令是 import，如要调用上述包，可用

```
import math
import random
import numpy
import scipy
import pandas
import matplotlib
```

这些包中的函数，可直接使用包名加“.”。如要用 matplotlib 绘 plot 图，可用 matplotlib.plot(...)。

如要简化这些包的写法，可用 as 命令赋予别名，如

```
import numpy as np
import scipy as sp
```

```
import pandas as pd
import matplotlib as plt
```

这样 matplotlib.plot(...)可简化为 plt.plot(...)。

如要调用 Python 包中某个具体函数或方法，可使用 from ... import，例如，要调用 math 包中的开方、对数和 pi 函数，则用

```
from math import sqrt, log, pi
```

这样，可在程序中直接使用，如 sqrt(2)，等价于 math.sqrt(2)。

比如，要调用本书自定义函数文档 PyDm_fun.py 中的函数（见相关章节及附录），需按如下方式操作：

(1)安装自定义模块：将 PyDm_fun.py 文档复制到当前工作目录 D:\PyDm 下。

(2)加载自定义模块：import PyDm_fun as dm　# from PyDm_fun import *

(3)自定义函数调用：dm.mcor_test（X）# mcor_test（X）

In [2]	# 系统初始化	
	import numpy as np	#加载数值分析包
	np.set_printoptions(precision=4)	#设置 numpy 输出精度
	import pandas as pd	#加载数据操作包
	pd.set_option('display.width', 120)	#设置 pandas 输出宽度
	pd.set_option('display.precision',4)	#设置 pandas 输出精度
	import matplotlib.pyplot as plt	#加载基本绘图包
	plt.rcParams['font.sans-serif']=['KaiTi'];	#SimHei 黑体
	plt.rcParams['axes.unicode_minus']=False;	#正常显示图中负号
	import PyDm_fun as dm	#from PyDm_fun import * #加载自定义函数

1.3.1.3　Python 中的数据管理

目前，Python 最大的问题是数据管理，因为 Python 没有好用的数据管理器，其自带的数据管理器很不方便，所以，要用好 Python 软件，就得将 Python 与 Excel 等电子表格充分结合，发挥两者的优点，这样就可以事半功倍，这也是本书提出用“电子表格+Python”模式进行数据统计分析的原因。关于如何使用 Python 调用 Excel 等电子表格数据，参见 1.3.4.3 节。

1.3.2　Python 数据类型

1.3.2.1　Python 对象

Python 创建和控制的实体称为对象(object)，它们可以是变量、数组、字符串、函数或结构。由于 Python 是一种所见即所得的脚本语言，故不需要编译。在 Python 里，对象是通过名字创建和保存的。可以用 who 命令来查看当前打开的 Python 环境里的对象，用 del 删除这些对象。

(1)查看数据对象

In [3]	who
Out [3]	dm np pd

(2)生成数据对象

In [4]	x=10.12 #创建对象 x who
Out [4]	dm np pd x

(3)删除数据对象

In [5]	del x #删除对象 x who
Out [5]	dm np pd

上面列出的是新创建的数据对象 x 的名称。Python 对象的名称必须以一个英文字母打头，并由一串大小写字母、数字或下画线组成。**注意**：Python 区分大小写，比如，Orange 与 orange 数据对象是不同的。不要用 Python 的内置函数名作为对象的名称，如 who、del 等。

1.3.2.2 数据的基本类型

Python 的基本数据类型包括数值型、逻辑型、字符型、复数型等，也可能是缺失值。

(1)数值型

数值型数据的形式是实数，可以写成整数(如 n=3)、小数(如 x=1.46)、科学计数(y=1e9)的方式，该类型数据默认是双精度数据。

Python 支持 4 种不同的数字类型：

int(有符号整型)；

long(长整型，也可以代表八进制和十六进制)；

float(浮点型)；

complex(复数)。

说明：Python 中显示数据或对象内容直接用其名称，相当于执行 print 函数，见下。

In [6]	n=10 #整数
	n #无格式输出，相当于 print(n)
	print("n=",n) #有格式输出
	x=10.234 #实数
	print(x)
	print("x=%10.5f"%x)

Out [6]	10 n= 10 10.234 x= 10.23400

(2)逻辑型

逻辑型数据只能取值 True 或 False。

In [7]	a=True;a b=False;b
Out [7]	True False

可以通过比较获得逻辑型数据，如下所示。

In [8]	10>3 10<3
Out [8]	True False

(3)字符型

字符型数据的形式是夹在双引号" "或单引号' '之间的字符串，如'MR'。**注意**：一定要用英文引号，不能用中文引号“ ”或‘ ’。Python 语言中的 string(字符串)是由数字、字母、下画线组成的一串字符。一般形式为

```
s = 'I love Python'
```

它是编程语言中表示文本的数据类型。

另外，Python 字符串具有切片功能，即从左到右索引默认从 0 开始，最大范围是字符串长度减一(左闭右开)；从右到左索引默认从–1 开始。如果要实现从字符串中获取一段子字符串，可以使用变量[头下标:尾下标]，其中下标从 0 开始算起，可以是正数或负数，也可以为空，表示取到头或尾。比如，上例中 s[7]的值是 p，s[2:6]的结果是 love。

加号(+)是字符串连接运算符，星号(*)是重复操作。

In [9]	s = 'We love Python';s s[7] s[2:6] s+' '+s s*2
Out [9]	'We love Python' 'P' 'love' 'We love Python We love Python' 'We love PythonWe love Python'

(4)缺失值

有些统计资料是不完整的。当一个元素或值在统计的时候是“不可得到”或“缺失值”的时候，相关位置可能会被保留并且赋予一个特定的 nan(not available number，不是一个数)值。任何 nan 的运算结果都是 nan。如 float('nan')就是一个实数缺失值。

(5)数据类型转换

有时，需要对数据内置的类型进行转换。数据类型的转换，只须将数据类型作为函数名即可。以下几个内置的函数可以实现数据类型之间的转换。这些函数返回一个新的对象，表示转换的值。下面列出几种常用的数据类型转换方式：

```
int(x [,base])        #将 x 转换为一个整数
float(x)              #将 x 转换为一个浮点数
str(x)                #将对象 x 转换为字符串
chr(x)                #将一个整数转换为一个字符
```

Python 的所有数据类型都是类，可以通过 type( )函数查看该变量的数据类型。

1.3.2.3 标准数据类型

在内存中存储的数据可以有多种类型。例如，一个人的年龄可以用数字来存储，名字可以用字符来存储。Python 定义了一些标准类型，用于存储各种类型的数据。这些标准的数据类型是由前述基本类型构成的。

(1) list(列表)

list(列表)是 Python 中使用最频繁的数据类型。列表可以完成大多数集合类的数据结构实现。它支持字符、数字、字符串，甚至可以包含列表(即嵌套)。列表用 [] 标识，是一种最通用的复合数据类型。Python 的列表也具有切片功能，列表中值的切割也可以用到变量 [头下标:尾下标]，可以截取相应的列表，从左到右索引默认从 0 开始，从右到左索引默认从–1 开始，下标可以为空，表示取到头或尾。

加号 + 是列表连接运算符，星号 * 是重复操作。操作方法类似字符串。

列表 list 是进行数据分析的基本类型，所以必须掌握。

In [10]	list1 =[] #空列表 list1 list1 = ['Python', 786 , 2.23, 'R', 70.2] list1 #输出完整列表 list1[0] #输出列表的第一个元素 list1[1:3] #输出第二个至第三个元素 list1[2:] #输出从第三个开始至列表末尾的所有元素 list1 * 2 #输出列表两次 list1 + list1[2:4] #打印组合的列表
Out [10]	[] ['Python', 786, 2.23, 'R', 70.2] 'Python' [786, 2.23] [2.23, 'R', 70.2] ['Python', 786, 2.23, 'R', 70.2, 'Python', 786, 2.23, 'R', 70.2] ['Python', 786, 2.23, 'R', 70.2, 2.23, 'R']
In [11]	X=[1,3,6,4,9]; X sex=['女','男','男','女','男'] sex weight=[67,66,83,68,70]; weight
Out [11]	[1, 3, 6, 4, 9] ['女', '男', '男', '女', '男'] [67, 66, 83, 68, 70]

(2) tuple(元组)

元组是另一种数据类型，类似于 list(列表)。元组用"()"标识，内部元素用逗号隔开。元组不能赋值，相当于只读列表。操作类似列表。

(3) dictionary(字典)

字典也是一种数据类型，且可存储任意类型对象。字典的每个键值对用冒号“:”分隔，每个键值对之间用逗号“,”分隔，整个字典包括在花括号{}中，格式如下：

```
dict= {key1 : value1, key2 : value2 }
```

键必须是唯一的，但值则不必，值可以取任何数据类型，如字符串、数字或元组。

字典是除列表外 Python 中最灵活的内置数据结构类型。列表是有序的对象集合，字典是无序的对象集合。

两者之间的区别在于：字典中的元素是通过键来存取的，而不是通过下标存取。

In [12]	{} dict1={'name':'john','code':6734,'dept':'sales'};dict1 dict1['code'] dict1.keys() dict1.values()	#空字典 #定义字典 #输出键为'code' 的值 #输出所有键 #输出所有值
Out [12]	{'code': 6734, 'dept': 'sales', 'name': 'john'} 6734 dict_keys(['name', 'code', 'dept']) dict_values(['john', 6734, 'sales'])	
In [13]	dict2={'sex': sex,'weight':weight}; dict2	#根据列表构成字典
Out [13]	{'sex': ['女', '男', '男', '女', '男'], 'weight': [67, 66, 83, 68, 70]}	

1.3.3 数值分析包 numpy

在使用 numpy 包前，须加载其到内存中，语句为 import numpy，通常将其简化为

```
import numpy as np
```

1.3.3.1 一维数组（向量）

下面是使用 Python 的 numpy 包对一维数组或向量的基本操作。

In [14]	import numpy as np np.array([1,2,3,4,5])	#加载数组包 #一维数组
Out [14]	array([1, 2, 3, 4, 5])	
In [15]	np.array([1,2,3,np.nan,5])	#包含缺失值的数组
Out [15]	array([1., 2., 3., nan, 5.])	
In [16]	np.arange(9) np.arange(1,9,0.5) np.linspace(1,9,5)	#数组序列 #等差数列 #等距数列
Out [16]	array([0, 1, 2, 3, 4, 5, 6, 7, 8]) array([1. , 1.5, 2. , 2.5, 3. , 3.5, 4. , 4.5, 5. , 5.5, 6. , 6.5, 7. , 7.5, 8. , 8.5]) array([1., 3., 5., 7., 9.])	

1.3.3.2 二维数组（矩阵）

下面是使用 Python 的 numpy 包构建二维数组或矩阵的基本函数。

In [17]	np.array([[1,2],[3,4],[5,6]])	#二维数组
Out [17]	array([[1, 2], [3, 4], [5, 6]])	
In [18]	A=np.arange(9).reshape((3,3));A	#形成 3×3 矩阵

Out [18]	array([[0, 1, 2], [3, 4, 5], [6, 7, 8]])

1.3.3.3 数组的操作

下面是对数组操作的一些常用函数。

(1) 数组的维度

In [19]	A.shape
Out [19]	(3, 3) #元组类型

(2) 空数组

In [20]	np.empty([3,3]) #空数组
Out [20]	array([[0.00000000e+000, 9.56511090e−321, 2.47032823e−323], [1.13844130e−311, 0.00000000e+000, 2.47032823e−323], [1.13844129e−311, 0.00000000e+000, 3.23815565e−319]])

(3) 零数组

In [21]	np.zeros((3,3)) #零矩阵
Out [21]	array([[0., 0., 0.], [0., 0., 0.], [0., 0., 0.]])

(4) 1 数组

In [22]	np.ones((3,3)) #1 矩阵
Out [22]	array([[1., 1., 1.], [1., 1., 1.], [1., 1., 1.]])

(5) 单位阵

In [23]	np.eye(3) #单位阵
Out [23]	array([[1., 0., 0.], [0., 1., 0.], [0., 0., 1.]])

1.3.4 数据分析包 pandas

在数据分析中，数据通常以变量(一维数组，Python 中用序列表示)和矩阵(二维数组，Python 中用数据框表示)的形式出现，下面结合 Python 介绍 pandas 基本的数据操作。

注意：在 Python 编程中，变量通常以列表(一组数据)，而不是一般编程语言的标量(一个数据)形式出现。

1.3.4.1 序列 Series

(1) 创建序列(向量、一维数组)

假如要创建一个含有 n 个数值的向量 $X=(x_1,x_2,\ldots,x_n)$，Python 中创建序列的函数是列表，这些向量可以是数字型的，也可以是字符串型的，还可以是混合型的。

特别**说明**：Python 中显示数据或对象内容直接用其名称，见下。

(2) 生成序列

In [24]	import pandas as pd　　#加载数据分析包 pd.Series()　　#生成空序列
Out [24]	Series([], dtype: float64)

(3) 根据列表构建序列

In [25]	X=[1,3,6,4,9] S1=pd.Series(X);S1 S2=pd.Series(weight);S2 S3=pd.Series(sex);S3
Out [25]	…… 0　女 1　男 2　男 3　女 4　男

(4) 序列合并

In [26]	pd.concat([S2,S3],axis=0)　　#按行合并序列 pd.concat([S2,S3],axis=1)　　#按列合并序列
Out [26]	0　67 1　66 2　83 3　68 4　70 0　女 1　男 2　男 3　女 4　男 　0　1 0　67　女 1　66　男 2　83　男 3　68　女 4　70　男

(5) 序列切片

In [27]	S1[2] S3[1:4]
Out [27]	6 1　男 2　男 3　女

1.3.4.2　数据框 DataFrame

pandas 中的函数 DataFrame() 可用序列构成一个数据框，如下页的 df1 和 df2。数据框相

当于关系数据库中的结构化数据类型，传统的数据大都以结构化数据形式存储于关系数据库中，因而传统的数据分析是以数据框为基础的。Python 中的数据分析大都是基于数据框进行的，所以本书的分析也是以该数据类型为主，向量和矩阵都可以看成数据框的一个特例。

(1) 生成数据框

In [28]	pd.DataFrame()　　#生成空数据框
Out [28]	Empty DataFrame Columns: [] Index: []

(2) 根据列表创建数据框

In [29]	pd.DataFrame(X) pd.DataFrame(X, columns=['X'], index=range(5)) pd.DataFrame(weight,columns=['weight'], index=['A','B','C','D','E'])
Out [29]	…… weight A 67 B 66 C 83 D 68 E 70

(3) 根据字典创建数据框

In [30]	df1=pd.DataFrame({'S1':S1,'S2':S2,'S3':S3}); df1
Out [30]	S1 S2 S3 0 1.0 67 女 1 3.0 66 男 2 6.0 83 男 3 4.0 68 女 4 9.0 70 男
In [31]	df2=pd.DataFrame({'sex':sex,'weight':weight},index=X);df2
Out [31]	sex weight 1 女 67 3 男 66 6 男 83 4 女 68 9 男 70

(4) 增加数据框列

In [32]	df2['weight2']=df2['weight']**2; df2　　#生成新列
Out [32]	sex weight weight2 1 女 67 4489 3 男 66 4356 6 男 83 6889 4 女 68 4624 9 男 70 4900

(5) 删除数据框列

In [33]	del df2['weight2']; df2　　#删除数据列

Out [33]	sex weight 1 女 67 3 男 66 6 男 83 4 女 68 9 男 70

(6)缺失值处理

In [34]	df3=pd.DataFrame({'S2':S2,'S3':S3},index=S1);df3	
Out [34]	S2 S3 1 66.0 男 3 68.0 女 6 NaN NaN 4 70.0 男 9 NaN NaN	
In [35]	df3.isnull()	#若是缺失值则返回 True，否则返回 False
Out [35]	S2 S3 1 False False 3 False False 6 True True 4 False False 9 True True	
In [36]	df3.isnull().sum()	#返回每列包含的缺失值的个数
Out [36]	S2 2 S3 2	
In [37]	df3.dropna()	#直接删除含有缺失值的行，多变量谨慎使用
Out [37]	S2 S3 1 66.0 男 3 68.0 女 4 70.0 男	

(7)数据框排序

In [38]	df3.sort_index()	#按 index 排序
Out [38]	S2 S3 1 66.0 男 3 68.0 女 4 70.0 男 6 NaN NaN 9 NaN NaN	
In [39]	df3.sort_values(by='S3')	#按列值排序
Out [39]	S2 S3 3 68.0 女 1 66.0 男 4 70.0 男 6 NaN NaN 9 NaN NaN	

1.3.4.3 数据框的读写

(1)pandas 读取数据集

大的数据对象常常是从外部文件读入，而不是在 Python 中直接输入的。外部的数据源有很多，可以是电子表格、数据库、文本文件等形式。Python 的导入工具非常简单，但是对导入文件有一些比较严格的限制。本书使用的是 pandas 包读取数据的方式，事先须调用 pandas 包，即 import pandas。

① 从剪贴板上读取。

前面讲到，电子表格是目前数据管理和编辑最方便的工具，所以可以考虑用电子表格管理数据，用 Python 分析数据，电子表格与 Python 之间的数据交换(适用于全书)过程非常简单，简述如下。

先在 PyDm_data.xls 数据文件的【BSdata】表中选取 A1:H52，复制，然后在 Python 中读取数据。

```
In [40]: BSdata=pd.read_clipboard();    #从剪贴板上复制数据
         BSdata[:5]                     #BSdata.head() 见下一节
```

Out [40]

	学号	性别	身高	体重	支出	开设	课程	软件
0	1510248008	女	167	71	46.0	不清楚	都未学过	No
1	1510229019	男	171	68	10.4	有必要	概率统计	Matlab
2	1512108019	女	175	73	21.0	有必要	统计方法	SPSS
3	1512332010	男	169	74	4.9	有必要	编程技术	Excel
4	1512331015	男	154	55	25.9	有必要	都学习过	Python

这里，BSdata 为读入 Python 中的数据框名，clipboard 为剪贴板。

② 读取 csv 格式数据。

虽然 Python 可以直接复制表格数据，但也可读取电子表格工作簿中的一个表格(例如，在 Excel 中将数据 DaPy_data.xlsx 的表单[BSdata]另存为 BSdata.csv，这时 BSdata.csv 本质上也是文本文件，是以逗号分隔的文本数据，既可用记事本打开，也可用电子表格软件打开，是最通用的数据格式)，其读取命令也最简单，如下所示。

```
In [41]: BSdata=pd.read_csv("BSdata.csv",encoding='utf-8')  #注意中文格式
         BSdata[6:9]
```

Out [41]

	学号	性别	身高	体重	支出	开设	课程	软件
6	1516352030	女	169	71	9.1	有必要	编程技术	Excel
7	1516171019	女	166	66	2.5	不必要	都未学过	Excel
8	1516391008	女	165	69	35.6	不必要	都未学过	Excel

③ 读取 Excel 格式数据。

使用 pandas 包中的 read_excel()函数可直接读取 Excel 文档中的任意表单数据，其读取命令也比较简单，例如，要读取 PyDm_data.xlsx 表单的[BSdata]，可用以下命令。

```
In [42]: BSdata=pd.read_excel('PyDm_data.xlsx','BSdata');BSdata[-5:]
```

Out [42]

	学号	性别	身高	体重	支出	开设	课程	软件
47	1538319004	男	175	68	44.4	不清楚	统计方法	SAS
48	1538254010	女	166	65	5.3	不清楚	编程技术	Python

	49	1540294017	女	159	58	71.4	不清楚	都学习过	SPSS
	50	1540365026	女	169	73	5.5	有必要	统计方法	Excel
	51	1540388036	女	165	67	56.8	不必要	概率统计	SAS

④ 读取其他统计软件的数据。

要调用 SAS、SPSS、Stata 等统计软件的数据集，须先用相应的包，详见 Python 手册。

(2) pandas 数据集的保存

Python 读取和保存数据集的最好方式是 csv 和 xlsx 文件格式，pandas 保存数据的命令也很简单，如下所示。

```
In [43]: BSdata.to_csv('BSdata1.csv')                    #将数据框 BSdata 保存到 BSdata.csv 中
         BSdata.to_excel('BSdata1.xlsx',index=False)     #将数据框 BSdata 保存到 BSdata1.xlsx 中
```

1.3.4.4 数据框的操作

(1) 获取数据框的基本信息

① 数据框显示。

有三种显示数据框内容的函数，即 info()(显示数据结构)、head()(显示数据框前 5 行)、tail()(显示数据框后 5 行)。

```
In [44]: BSdata.info()                                   #数据框信息
Out [44]: <class 'pandas.core.frame.DataFrame'>
          RangeIndex: 52 entries, 0 to 51
          Data columns (total 8 columns):
          学号      52 non-null int64
          性别      52 non-null object
          身高      52 non-null int64
          体重      52 non-null int64
          支出      52 non-null float64
          开设      52 non-null object
          课程      52 non-null object
          软件      52 non-null object
          dtypes: float64(1), int64(3), object(4)
          memory usage: 3.3+ KB
In [45]: BSdata.head()                                   #显示前 5 行
         BSdata.tail()                                   #显示后 5 行
Out [45]:
```

	学号	性别	身高	体重	支出	开设	课程	软件
0	1510248008	女	167	71	46.0	不清楚	都未学过	No
1	1510229019	男	171	68	10.4	有必要	概率统计	Matlab
2	1512108019	女	175	73	21.0	有必要	统计方法	SPSS
3	1512332010	男	169	74	4.9	有必要	编程技术	Excel
4	1512331015	男	154	55	25.9	有必要	都学习过	Python
	学号	性别	身高	体重	支出	开设	课程	软件
47	1538319004	男	175	68	44.4	不清楚	统计方法	SAS
48	1538254010	女	166	65	5.3	不清楚	编程技术	Python
49	1540294017	女	159	58	71.4	不清楚	都学习过	SPSS
50	1540365026	女	169	73	5.5	有必要	统计方法	Excel
51	1540388036	女	165	67	56.8	不必要	概率统计	SAS

② 数据框列名(变量名)。

In [46]	BSdata.columns　　#查看列名
Out [46]	Index(['学号', '性别', '身高', '体重', '支出', '开设', '课程', '软件'], dtype='object')

③ 数据框行名(样品名)。

In [47]	BSdata.index　　#数据框行名
Out [47]	RangeIndex(start=0, stop=52, step=1)

④ 数据框维度。

In [48]	BSdata.shape　　#显示数据框的行数和列数 BSdata.shape[0]　　#数据框行数 BSdata.shape[1]　　#数据框列数
Out [48]	(52, 8) 52 8

⑤ 数据框值(数组)。

In [49]	BSdata.values[:5]　　#数据框值数组
Out [49]	array([[1510248008, '女', 167, 71, 46.0, '不清楚', '都未学过', 'No'], [1510229019, '男', 171, 68, 10.4, '有必要', '概率统计', 'Matlab'], [1512108019, '女', 175, 73, 21.0, '有必要', '统计方法', 'SPSS'], [1512332010, '男', 169, 74, 4.9, '有必要', '编程技术', 'Excel'], [1512331015, '男', 154, 55, 25.9, '有必要', '都学习过', 'Python']], dtype=object)

(2)选取变量

选取数据框中变量的方法主要有以下几种。

① “.”法或[' ']：这是 Python 中最直观的选取变量的方法，比如，要选取数据框 BSdata 中的“身高”和“体重”变量，直接用“BSdata.身高”与“BSdata.体重”即可，也可用 BSdata['身高']与 BSdata['体重']，该方法书写比“.”法烦琐，却是最不容易出错且直观的一种方法，并可推广到多个变量的情形，推荐使用。

In [50]	BSdata.身高　　#取一列数据，BSdata['身高']
Out [50]	0　　167 1　　171 2　　175 3　　169 4　　154 .....
In [51]	BSdata[['身高','体重']]　　#取两列数据
Out [51]	身高　体重 0　167　71 1　171　68

	2 175 73 3 169 74 4 154 55 .....

② 下标法：由于数据框是二维数组(矩阵)的扩展，所以也可以用矩阵的列下标来选取变量数据，这种方法进行矩阵(数据框)运算比较方便。例如，dat.iloc[i,j]表示数据框(矩阵)的第 i 行、第 j 列数据，dat.iloc[i,]表示 dat 的第 i 行数据向量，而 dat.iloc[,j]表示 dat 的第 j 列数据向量(变量)。再如，“身高”和“体重”变量在数据框 BSdata 的第 3、4 两列。

In [52]	BSdata.iloc[:,2] #取 1 列 BSdata.iloc[:,2:4] #取 3、4 列
Out [52]	0 167 1 171 2 175 3 169 4 154 ..... 身高 体重 0 167 71 1 171 68 2 175 73 3 169 74 4 154 55

(3)提取样品

In [53]	BSdata.loc[3] #取 1 行
Out [53]	学号 1512332010 性别 男 身高 169 体重 74 支出 4.9 开设 有必要 课程 编程技术 软件 Excel
In [54]	BSdata.loc[3:5] #取 3 至 5 行
Out [54]	学号 性别 身高 体重 支出 开设 课程 软件 3 1512332010 男 169 74 4.9 有必要 编程技术 Excel 4 1512331015 男 154 55 25.9 有必要 都学习过 Python 5 1516248014 男 183 76 85.6 不必要 编程技术 Excel

(4)选取观测与变量

同时选取观测与变量数据的方法就是将提取变量和样品方法结合使用。例如，要选取数据框中男生的身高数据，可用以下语句。

In [55]	BSdata.loc[:3,['身高','体重']] BSdata.iloc[:3,:5] #0 至 2 行和 1 至 5 列数据

Out [55]	

```
   身高  体重
0  167   71
1  171   68
2  175   73
3  169   74
        学号  性别  身高  体重  支出
0  1510248008   女   167   71  46.0
1  1510229019   男   171   68  10.4
2  1512108019   女   175   73  21.0
```

(5)根据条件选取样品与变量

例如，选取身高超过 180cm 的男生的数据，以及身高超过 180cm 且体重小于 80kg 的男生的数据，可用以下语句。

In [56]	BSdata[BSdata['身高']>180]

Out [56]	学号	性别	身高	体重	支出	开设	课程	软件
5	1516248014	男	183	76	85.6	不必要	编程技术	Excel
10	1520100029	男	184	82	10.3	有必要	都学习过	SAS
21	1525352033	男	185	83	5.1	有必要	都学习过	SPSS
32	1530243029	男	186	87	9.5	不必要	都未学过	No

In [57]	BSdata[(BSdata['身高']>180) & (BSdata['体重']<80)]

Out [57]	学号	性别	身高	体重	支出	开设	课程	软件
5	1516248014	男	183	76	85.6	不必要	编程技术	Excel

(6)数据框的运算

① 生成新的数据框。

可以通过选择变量名来生成新的数据框。

In [58]	BSdata['体重指数']=BSdata['体重']/(BSdata['身高']/100)**2 round(BSdata[:5],2)

Out [58]	学号	性别	身高	体重	支出	开设	课程	软件	体重指数
0	1510248008	女	167	71	46.0	不清楚	都未学过	No	25.46
1	1510229019	男	171	68	10.4	有必要	概率统计	Matlab	23.26
2	1512108019	女	175	73	21.0	有必要	统计方法	SPSS	23.84
3	1512332010	男	169	74	4.9	有必要	编程技术	Excel	25.91
4	1512331015	男	154	55	25.9	有必要	都学习过	Python	23.19

② 数据框的合并 pd.concat()。

可以用 pd.concat()函数将两个或两个以上的向量、矩阵或数据框合并起来，参数 axis=0 表示按行合并，axis=1 表示按列合并。

- 按行合并，axis=0。

In [59]	pd.concat([BSdata.身高, BSdata.体重],axis=0)

```
Out [59]  0    167
          1    171
          2    175
```

	3　　169 4　　154 ……

- 按列合并，axis=1。

In　[60]	pd.concat([BSdata.身高, BSdata.体重],axis=1)
Out [60]	身高　体重 0　167　71 1　171　68 2　175　73 3　169　74 4　154　55 .....

(7)数据框转置(.T)

In　[61]	BSdata.iloc[:3,:5].T			
Out [61]		0	1	2
	学号	1510248008	1510229019	1512108019
	性别	女	男	女
	身高	167	171	175
	体重	71	68	73
	支出	46	10.4	21

1.3.5　Python 编程运算

1.3.5.1　基本运算

与 Basic 语言、VB 语言、C 语言、C++语言等一样，Python 语言具有编程功能，但 Python 是新时期的编程语言，具有面向对象的功能，同时 Python 还是面向函数的语言。既然 Python 是一种编程语言，它就具有常规语言的算术运算符和逻辑运算符(如表 1-5 所示)，以及控制语句、自定义函数等功能。下面对 Python 的编程特点做一些简单介绍。

表 1-5　Python 中常用的算术运算符和逻辑运算符

算术运算符	含　义	逻辑运算符	含　义
+	加	<(<=)	小于(小于等于)
-	减	>(>=)	大于(大于等于)
*	乘	==	等于
/	除	!=	不等于
**	幂	not x	非 x
%	取模	or	或
//	整除	and	与

1.3.5.2 控制语句

编程离不开对程序的控制，下面介绍几个最常用的控制语句，其他控制语句见 Python 手册。

(1) 循环语句 for

Python 的 for 循环可以遍历任何序列的项目，如一个列表或一个字符串。for 循环允许循环使用向量或数列的每个值，在编程时非常有用。

for 循环的语法格式如下：

```
for iterating_var in sequence:
    statements(s)
```

Python 的 for 循环比其他语言更为强大，例如：

In [62]	for i in range(1,5): print(i)
Out [62]	1 2 3 4
In [63]	fruits = ['banana', 'apple', 'mango'] for fruit in fruits: print('当前水果 :', fruit)
Out [63]	当前水果 : banana 当前水果 : apple 当前水果 : mango
In [64]	for var in BSdata.columns: print(var)
Out [64]	学号 性别 身高 体重 支出 开设 课程 软件 体重指数

(2) 条件语句 if/else

if/else 语句是分支语句中的主要语句，其格式如下：

In [65]	a = –100
	if a < 100:
	    print("数值小于 100")
	else:
	    print("数值大于 100")

Out [65]	数值小于 100

Python 中有更简洁的形式来表达 ifelse 语句。

In [66]	−a if a<0 else a
Out [66]	100

注意：循环和条件等语句中要输出结果，请用 print()函数，这时只用变量名是无法显示结果的。

1.3.5.3 函数定义

在较复杂的计算问题中，有时一个任务可能需要重复多次，这时不妨自定义函数，这么做的好处是，函数内的变量名是局部的，即函数运行结束后它们不再保存到当前的工作空间，这就可以避免许多不必要的混淆和内存空间占用。Python 与其他统计软件的区别之一是，可以随时随地自定义函数，而且可以像使用 Python 的内置函数一样使用自定义的函数。

不同于 SAS、SPSS 等基于过程的统计软件，Python 进行数据分析是基于函数和面向对象的，所有 Python 的命令都是以函数形式出现的，比如读取文本数据的 read_clipboard()函数和读取 csv 数据文件的 read_csv()函数，以及建立序列的 Series()函数和构建数据框的 DataFrame()函数。由于 Python 是开源的，故所有函数使用者都可以查看其源代码。下面简单介绍 Python 的函数定义方法。定义函数的句法：

```
def 函数名(参数 1, 参数 2, …):
    函数体
  return
```

要学好 Python 数据分析，就必须掌握 Python 中的函数及其编程方法。表 1-6 所示是 Python 中常用的数学函数。

表 1-6 Python 中常用的数学函数

math 中的数学函数	含义(针对数值)	numpy 中的数学函数	含义(针对数组)
abs(x)	数值的绝对值	len(x)	数组中元素个数
sqrt(x)	数值的平方根	sum(x)	数组中元素求和
log(x)	数值的对数	prod(x)	数组中元素求积
exp(x)	数值的指数	min(x)	数组中元素最小值
round(x,n)	有效位数 n	max(x)	数组中元素最大值
sin(x),cos(x),…	三角函数	sort(x)	数组中元素排序
		rank(x)	数组中元素秩次

函数名可以是任意字符，但之前定义过的要小心使用，后定义的函数会覆盖先定义的函数。

注意：如果函数只用来计算，不需要返回结果，则可在函数中用 print()函数，这时只用变量名是无法显示结果的，见下。

一旦定义了函数名，就可以像 Python 的其他函数一样使用，比如，定义一个用来求一组数据的均值的函数，可以用与 C、C++、VB 等语言相同的方式定义，但方便得多。如计算向量 $\boldsymbol{X}=(x_1, x_2,\cdots,x_n)$ 的均值函数：

$$\overline{x}=\frac{\sum_{i=1}^{n}x_i}{n}$$

代码如下：

In [67]	x=[1,3,6,4,9,7,5,8,2]; x def xbar(x): 　　n=len(x) 　　xm=sum(x)/n 　　return(xm) xbar(x)
Out [67]	5.0

当然，Python 已内建这些函数命令，可直接使用，如下。

In [68]	np.mean(x)
Out [68]	5.0

要了解任何一个 Python 函数，使用 help()函数即可，例如，命令 help(sum)或?sum 将显示 sum()函数的使用帮助。

1.3.5.4 面向对象

Python 是一种面向对象的语言(一般使用者可不了解)。

前面介绍的序列(向量、一维数组)，数据框(矩阵、二维数组)都是 Python 的数据对象，各种 Python 函数也是对象。由于 Python 函数的许多计算结果都放在对象中，这使得 Python 的结果通常比 SAS、SPSS 和 Stata 等数据分析软件的结果简洁，需要时才调用，为进一步分析提供了方便。

下面就通过编写一个函数的过程来简单介绍 Python 的函数自定义方法和面向对象技术。如计算向量 $\boldsymbol{X}=(x_1,x_2,\ldots,x_n)$ 的离均差平方和函数：

$$\mathrm{SS}=\sum_{i=1}^{n}(x_i-\overline{x})^2=\sum_{i=1}^{n}x_i^2-\left(\sum_{i=1}^{n}x_i\right)^2\Big/n$$

有了离均差平方和函数，就可做许多统计计算，如计算方差、标准差，进行方差分析等。该函数用来计算一些常用的统计量协方差阵和相关系数阵。

In [69]	X=np.array([1,3,6,4,9,7,5,8,2]);X　　#列表数组 def SS1(x): 　　n=len(x) 　　ss=sum(x**2)−sum(x)**2/n 　　return(ss) SS1(X)　　#SS1(BSdata.身高)
Out [69]	60.0

Python 一次可以返回多个数据对象，例如，下面的函数可返回数据的均值、平方和、离差平方和、方差、标准差，但一般要用到列表 list 类型。这里的列表是比数据框更高级的数据对象，相当于非结构化数据类型，列表类型为大数据分析提供了便利，其原因是，大数据中很多数据都呈非结构化特点。下面简单介绍 Python 列表类型的用法，初学者可暂不学习。

In [70]	def SS2(x): #返回多个值
	n=len(x)
	xm=sum(x)/n
	ss=sum(x**2)-sum(x)**2/n
	return[x**2,n,xm,ss] #return(x**2,n,xm,ss)
	SS2(X) #SS2(BSdata.身高)

Out [70]	[array([1, 9, 36, 16, 81, 49, 25, 64, 4], dtype=int32), 9, 5.0, 60.0]

向量、矩阵和数组的元素必须是同一类型的数据对象。如果一个数据对象需要包含不同类型的数据对象，可以采用列表的形式。列表是一个对象的有序集合，列表中包含的对象又称为它的成分，成分可以是不同的模式和类型，例如，一个列表可以包括数值向量、逻辑向量、矩阵、字符、数组和数据框等。

列表中对象的成分访问与变量和数据基本一样，可以用下标获取，但不完全一样，在此不详述。

In [71]	SS2(X)[0] #取第 1 个对象 SS2(X)[1] #取第 2 个对象 SS2(X)[2] #取第 3 个对象 SS2(X)[3] #取第 4 个对象
Out [71]	array([1, 9, 36, 16, 81, 49, 25, 64, 4], dtype=int32) 9 5.0 60.0

可以使用 type() 函数来查看数据或对象的类型。

In [72]	type(SS2(X))
Out [72]	list
In [73]	type(SS2(X)[3])
Out [73]	numpy.float64

数据及练习 1

1.1 下面有三组数据：

```
1, 2, 3, 4, 5
a, b, c, d
```

```
physics, chemistry, 1997, 2000
```

(1) 将其写入列表。

(2) 将其写入字典。

1.2 请创建下列 Python 数组，并计算。

(1) 创建一个 2*2 的数组，计算对角线上元素的和。

(2) 创建一个长度为 9 的一维数据，数组元素为 0～8，并将它重新变为 3*3 的二维数组。

(3) 创建两个 3*3 的数组，分别将它们合并为 3*6、6*3 的数组后，拆分为 3 个数组。

1.3 文本数据。下面有一些文本数据：

```
name,physics,Python,math,english
Google,100,100,25,12
Facebook,45,54,44,88
Twitter,54,76,13,91
Yahoo,54,452,26,100
```

(1) 请将其写入列表。

(2) 请将其写入字典。

(3) 请将其写入数据框。

(4) 请将其保存到 csv 格式的文档，并从 read_csv() 函数读入 Python。

1.4 调查数据。某公司对财务部门人员的抽烟状况进行调查，结果为：否，否，否，是，是，否，否，是，否，是，否，否，是，是，否，是，否，否，是，是。

(1) 请用列表录入该数据。

(2) 请将这组数据输入电子表格，并将其读入 Python。

1.5 学生成绩。从某大学统计系的学生中随机抽取 24 人，对数学和统计学的考试成绩进行调查，数据如表 1-7 所列。

表 1-7 部分学生的数学和统计学考试成绩

编号	性别	数学	统计学	编号	性别	数学	统计学
1	M	81	72	13	F	83	78
2	F	90	90	14	F	81	94
3	F	91	96	15	M	77	73
4	M	74	68	16	M	60	66
5	F	70	82	17	F	66	58
6	F	73	78	18	M	84	87
7	M	88	89	19	F	80	86
8	M	78	82	20	F	85	84
9	M	95	96	21	M	70	82
10	F	63	75	22	M	54	56

续表

编号	性别	数学	统计学	编号	性别	数学	统计学
11	F	85	86	23	F	93	98
12	M	60	71	24	M	68	76

(1) 试将这组数据输入电子表格。

(2) 分别用 Python 的 read_csv() 和 read_excel() 函数读取。

(3) 用 Python 方法获取性别、数学和统计学成绩变量，并筛选不同性别学生的成绩。

(4) 请在电子表格和 Python 中分别对性别、数学或统计学成绩排序。

1.6 电子表格。将上述所有数据统一放入一个 Excel 或 WPS 电子表格，每个 sheet 放一组，并给文档起名为 mydata1.xlsx，以备后用。

第 2 章　数据挖掘的分析基础

我们在进行任何统计分析之前，都需要对数据进行探索性分析(Exploratory Data Analysis，EDA)，以了解资料的性质和数据的特点。当面对一组陌生的数据时，进行探索性统计分析有助于我们掌握数据的基本情况。探索性数据分析是通过分析数据集来决定选择哪种方法进行统计推断的过程。对于一维数据，人们想知道数据是否近似地服从正态分布，是否呈现拖尾或截尾分布；它的分布是对称的，还是呈偏态的；分布是单峰、双峰的，还是多峰的；这一分析主要通过计算基本统计量和绘制基本统计图来实现。

2.1　数据的描述分析

2.1.1　基本统计量

Python 提供了很多对数据进行基本分析的函数，表 2-1 所列是 Python 对变量(序列或数据框)进行基本统计分析的函数。描述统计量函数 describe()可对数据做一基本描述，默认是分析计量数据的基本统计量。

In [1]
```
BSdata.describe()
```

Out [1]

	学号	身高	体重	支出
count	5.200000e+01	52.000000	52.000000	52.000000
mean	1.523270e+09	168.519231	68.500000	24.511538
std	1.899525e+07	8.018338	7.711718	21.432060
min	1.438120e+09	154.000000	50.000000	2.500000
25%	1.520377e+09	163.000000	63.000000	9.500000
50%	1.526685e+09	167.500000	68.500000	15.450000
75%	1.532229e+09	174.000000	73.000000	35.600000
max	1.540388e+09	186.000000	87.000000	85.600000

In [2]
```
BSdata[['性别','开设','课程','软件']].describe()
```

Out [2]

	性别	开设	课程	软件
count	52	52	52	52
unique	2	3	5	7
top	男	有必要	统计方法	Excel
freq	27	29	15	15

表 2-1　Python 对变量进行基本统计分析的函数

计数数据	用　途	计量数据	用　途
value_counts()	一维频数表	mean()	均值
crosstab()	二维列联表	median()	中位数
pivot_table()	多维透视表	quantile()	分位数
		std()	标准差

2.1.1.1　计数数据的汇总分析

统计学中把取值范围是有限个值或一个数列的变量称为离散变量，其中表示分类情况的数据又称为计数数据。

(1)频数：绝对数

Python 中的.value.counts()函数可对计数数据计算频数。

In [3]	T1=BSdata.性别.value_counts();T1
Out [3]	男　27 女　25

这是性别变量的频数分析，说明在 52 名学生中有男生 27 人、女生 25 人。

(2)频率：相对数

频数/总数为计数数据的频率。

In [4]	T1/sum(T1)*100
Out [4]	男　51.923077 女　48.076923

这是性别的频率分析，说明在 52 名学生中男生占 51.92%、女生占 48.08%。

2.1.1.2　计量数据的汇总分析

对于数值型数据，经常要分析它的集中趋势和离散程度。用来描述集中趋势的统计量主要有均值、中位数；描述离散程度的统计量主要有方差、标准差。Python 只需要一个函数就可以简单地得到这些结果。计算均值、中位数、方差、标准差的函数分别是 mean()、median()、var()、std()。

方差、标准差对异常值很敏感，可以用稳健的极差、四分位数间距(IQR)来描述离散程度。Python 还提供了函数 quantile()——对数据计算分位数，describe()——计算基本统计量。

计量数据的基本统计量主要包括均数、中位数、方差、标准差、极差和四分位数间距等，其基本含义如下。

(1)均数(算术平均数)

均数指一组数据的和除以这组数据的个数所得到的商，它反映了一组数据的总体水平。对于正态分布数据，通常计算其均数，来表示其集中趋势或平均水平。

$$\overline{X}=\frac{1}{n}\sum_{i=1}^{n}X_i$$

In [5]	BSdata.身高.mean()
Out [5]	168.51923076923077

(2) 中位数

中位数是将一组数据按大小顺序排列，处于中间位置的一个数据(或中间两个数据的平均值)，它反映了一组数据的集中趋势。对非正态分布数据，通常计算其中位数来表示其平均水平。

$$\overline{X}=\begin{cases} X_{\left(\frac{n+1}{2}\right)} & (n\text{为奇数}) \\ \frac{1}{2}\left[X_{\left(\frac{n}{2}\right)}+X_{\left(\frac{n}{2}+1\right)}\right] & (n\text{为偶数}) \end{cases}$$

In [6]	BSdata.身高.median()
Out [6]	167.5

(3) 极差

极差是一组数据中最大数据与最小数据的差，在统计中常用极差来刻画一组数据的离散程度。它反映的是变量分布的变异范围和离散幅度，在总体中任何两个单位的数值之差都不能超过极差。

$$R=X_{(n)}-X_{(1)}=\max(X)-\min(X)$$

In [7]	BSdata.身高.max()–BSdata.身高.min()
Out [7]	32.0

(4) 方差

方差是各数据与平均数之差的平方的均数，它表示数据离散程度和数据的波动大小。

$$s^2=\frac{1}{n-1}\sum_{i=1}^{n}(X_i-\overline{X})^2$$

In [8]	BSdata.身高.var()
Out [8]	64.29374057315236

(5) 标准差

标准差是方差的算术平方根，作用等同于方差，但单位与原数据单位是一致的。对正态分布数据，通常计算其标准差来反映其变异水平。

$$s=\sqrt{s^2}$$

In [9]	BSdata.身高.std()
Out [9]	8.01833776871194

方差或标准差是表示一组数据的波动性的指标，因此，通过方差或标准差可以

判断一组数据的稳定性：方差或标准差越大，数据越不稳定；方差或标准差越小，数据越稳定。

(6) 四分位数间距(IQR)

对非正态分布数据，通常计算其四分位数间距来反映其变异水平：IQR = Q_3–Q_1，其中，Q_3 和 Q_1 分别为数据的第 3 分位数和第 1 分位数(或称 75%和 25%分位数)。Python 提供了函数 quantile()，可对计量数据计算分位数，于是 IQR 可写为

$$\text{IQR} = \text{quantile}(x,0.75) - \text{quantile}(\text{x},0.25)$$

In [10]	BSdata.身高.quantile(0.75)–BSdata.身高.quantile(0.25)
Out [10]	11.0

(7) 偏度(skew)

偏度是描述数据分布偏斜方向和程度的度量，是统计数据分布非对称程度的数字特征。偏度亦称偏态、偏态系数，是表征概率分布密度曲线相对于均值不对称程度的特征数或特征量。直观看来就是密度函数曲线尾部的相对长度。

定义中偏度是样本的三阶标准化矩，定义式如下：

$$\text{skew} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^3 / n}{s^{3/2}}$$

In [11]	BSdata.身高.skew()
Out [11]	0.29880755120910174

(8) 峰度(kurt)

峰度与偏度类似，是描述总体中所有取值分布形态陡缓程度的统计量。该统计量需要与正态分布相比较，峰度为 0，表示该总体数据分布与正态分布的陡缓程度相同；峰度大于 0，表示该总体数据分布与正态分布相比较为陡峭，为尖顶峰；峰度小于 0，表示该总体数据分布与正态分布相比较为平坦，为平顶峰。峰度的绝对值数值越大，表示其分布形态的陡缓程度与正态分布的差异程度越大。

$$\text{kurt} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^4 / n}{s^2} - 3$$

In [12]	BSdata.身高.kurt()
Out [12]	–0.42072371559816935

Python 的特点是基于对象的函数分析，Python 中的所有分析工具都是基于函数的。要发挥 Python 的优势，通常可构建一些数据分析函数来进行基本的数据分析。

(9) 自编计算基本统计量函数

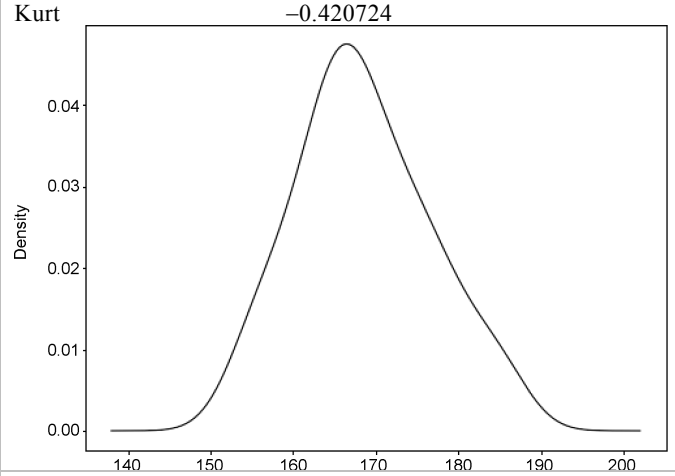
In [13]	def stats(x):

```
    stat=[x.count(),x.min(),x.quantile(.25),x.mean(),x.median(),
          x.quantile(.75),x.max(),x.max()-x.min(),x.var(),x.std(),x.skew(),x.kurt()]
    stat=pd.Series(stat,index=['Count','Min', 'Q1(25%)','Mean','Median',
                               'Q3(75%)','Max','Range','Var','Std','Skew','Kurt'])
    x.plot(kind='kde')     #拟合核密度 kde 曲线
    return(stat)
```

In [13]: stats(BSdata.身高)

Out [13]:

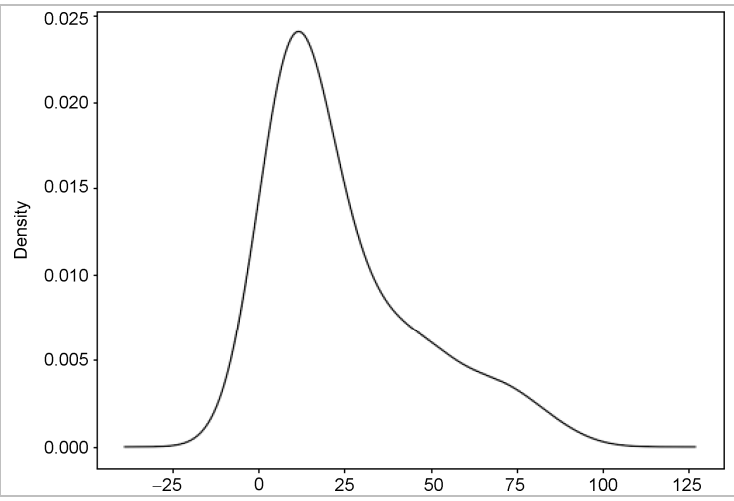
Count	52.000000
Min	154.000000
Q1(25%)	163.000000
Mean	168.519231
Median	167.500000
Q3(75%)	174.000000
Max	186.000000
Range	32.000000
Var	64.293741
Std	8.018338
Skew	0.298808
Kurt	−0.420724



In [14]: stats(BSdata.支出)

Out [14]:

Count	52.000000
Min	2.500000
Q1(25%)	9.500000
Mean	24.511538
Median	15.450000
Q3(75%)	35.600000
Max	85.600000
Range	83.100000
Var	459.333198
Std	21.432060
Skew	1.268351
Kurt	0.673127



当然，这些函数还可以不断完善，例如，它只能计算向量或变量数据，而无法计算矩阵或数据框的数据，用户可以自定义一个计算矩阵或数据框的基本统计量函数。

2.1.2 基本绘图函数

2.1.2.1 常用的绘图函数

matplotlib 是 Python 的基本绘图包，是 Python 的图形框架，类似于 Matlab 和 R 语言。它是 Python 中最著名的绘图包，提供了一整套和 Matlab 相似的命令 API，十分适合交互式地进行制图。在绘制中文图形时，需要做一些基本设置。

```
In  [15]  import matplotlib.pyplot as plt                    #基本绘图包
          plt.rcParams['font.sans-serif']=['KaiTi'];         #SimHei 黑体
          plt.rcParams['axes.unicode_minus']=False;          #正常显示图中负号
          plt.figure(figsize=(5,4));                         #图形大小
```

表 2-2 常用的绘图函数

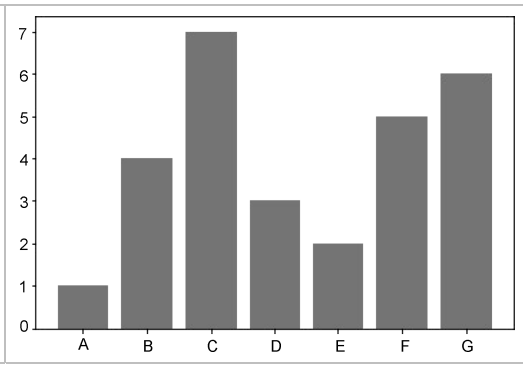
计数数据	用途	计量数据	用途
bar()	条图	plot()	线图
pie()	饼图	hist()	直方图

(1)计数数据的基本统计图

① 条图。

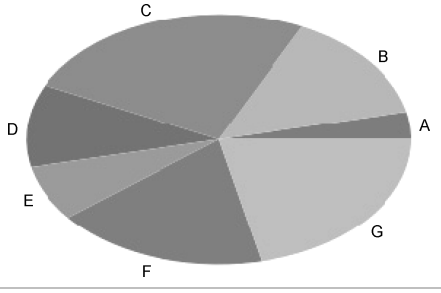
条图的高度可以是频数或频率，图的形状看起来一样，但是刻度不一样。matplotlib 画条图的函数是 bar()。在对分类数据作条图时，须先对原始数据分组，否则作出的不是分类数据的条图。

```
In  [16]  X=['A','B','C','D','E','F','G']
          Y=[1,4,7,3,2,5,6]
          plt.bar(X,Y);
```

Out [16]	

② 饼图。

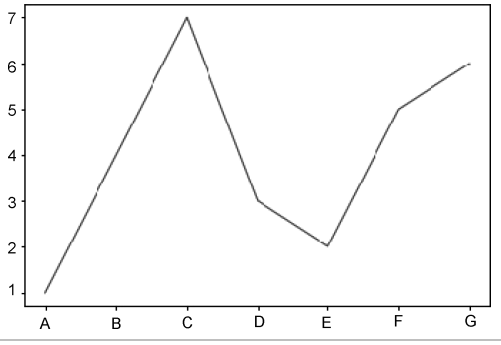
对分类数据还可以用饼图描述。饼图用于表示各类别的构成比情况，它以图形的总面积为 100%，扇形面积的大小表示事物内部各组成部分所占的百分比。在 matplotlib 中作饼图也很简单，只要使用函数 pie()就可以了。值得注意的是，和条图一样，对原始数据作饼图前要先分组。

In [17]	plt.pie(Y,labels=X);
Out [17]	

(2)计量数据的基本统计图

① 线图。

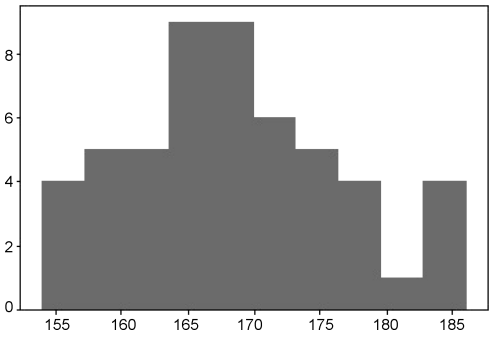
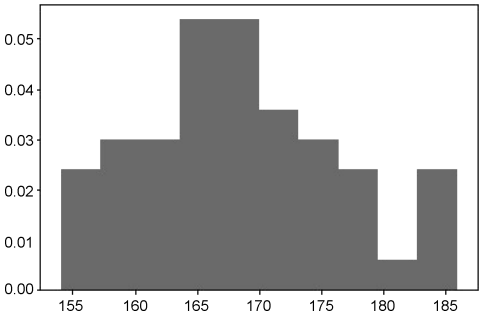
线图可以显示随时间而变化的连续数据，主要用于显示在相等时间间隔下数据的趋势。

In [18]	plt.plot(X,Y);
Out [18]	

② 直方图。

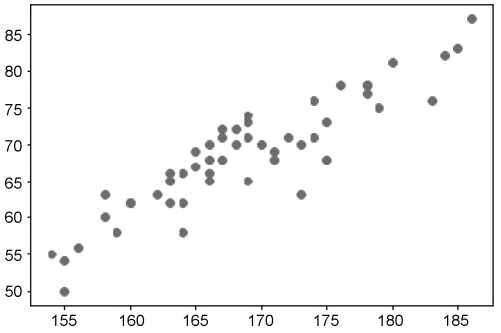
直方图用于表示连续型变量的频数分布，常用于考察变量的分布是否服从某种分布

类型，如正态分布。图形以矩形的面积表示各组段的频数(或频率)，各矩形的面积总和为总频数(或等于 1)。matplotlib 中用来作直方图的函数是 hist()，也可以用频率作直方图，只要把 density 参数设置为 True 就可以了，默认为 False。

In [19]	plt.hist(BSdata.身高)　　#频数直方图
Out [19]	(array([4., 5., 5., 9., 9., 6., 5., 4., 1., 4.]), array([154.,157.2,160.4, 163.6, 166.8,170.,173.2,176.4,179.6, 182.8,186.]), 
In [20]	plt.hist(BSdata.身高,density=True)　　#频率直方图
Out [20]	(array([0.02403846,0.03004808,0.03004808,0.05408654,0.05408654, 0.03605769, 0.03004808, 0.02403846, 0.00600962, 0.02403846]), array([154,157.2,160.4,163.6,166.8,170,173.2,176.4,179.6,182.8,186]) 

③散点图。

散点图表示一个变量随另一个变量变化的大致趋势，据此可以选择合适的函数对数据点进行拟合。

In [21]	plt.scatter(BSdata.身高, BSdata.体重);
Out [21]	

这些图是 Python 默认的形式，比较原始。可以通过设置不同的图形参数对图形进行调整和优化。

(3) 图形参数设置

Python 中的每个绘图函数，都有许多参数设置选项，大多数函数的部分选项是一样的，下面列出一些主要的共同选项及其缺失值。

① 标题、标签、标尺及颜色。

在使用 matplotlib 模块画坐标图时，往往需要对坐标轴设置很多参数，这些参数包括横/纵坐标轴范围、坐标轴刻度大小、坐标轴名称等。

在 matplotlib 中有很多函数，用来对这些参数进行设置，例如：

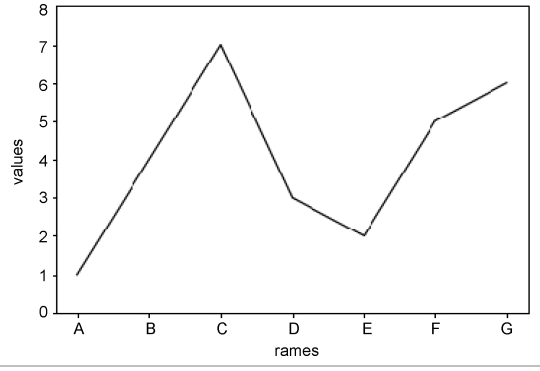
plt.xlim()，plt.ylim()：设置横/纵坐标轴范围；

plt.xlabel()，plt.ylabel()：设置坐标轴名称；

plt.xticks()，plt.yticks()：设置坐标轴刻度。

colors 参数用来控制图形的颜色，可简写为 c，c='red'表示设置为红色。

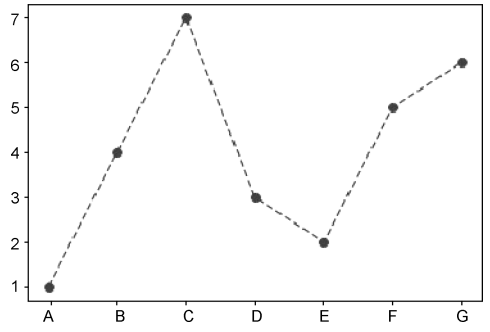
In [22]	plt.plot(X,Y,c='red');plt.ylim(0,8);
	plt.xlabel('names');plt.ylabel('values');
	plt.xticks(range(len(X)), X);

Out [22]	

② 线型和符号。

linestyle 参数用来控制连线的线型(-: 实线，--: 虚线，.: 点线)。

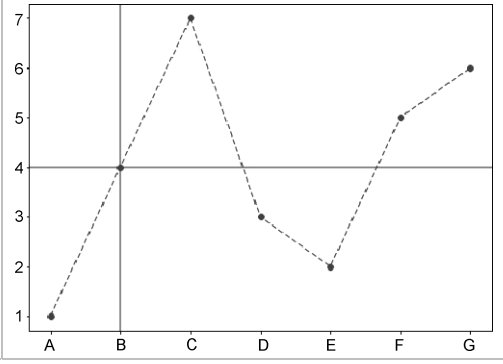
marker 参数用来控制符号的类型，例如，'o'为绘制实心圆点图。

In [23]	plt.plot(X,Y,linestyle='--',marker='o');
Out [23]	

③ 绘图函数附加图形。

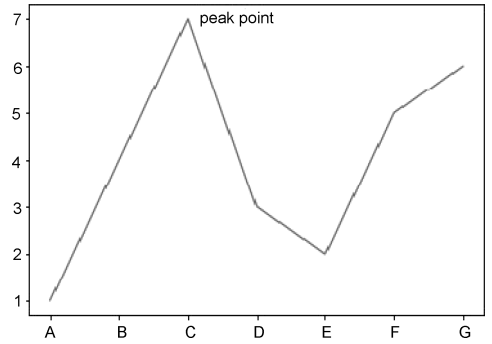
使用高级绘图函数可以画出一幅新图，而低级绘图函数只能作用于已有的图形之上。

- 垂线：在纵坐标 y 处画垂直线(plt.axvline())；
- 水平线：在横坐标 x 处画水平线(plt.axhline())。

In [24]	plt.plot(X,Y,'o--'); plt.axvline(x=1);plt.axhline(y=4);
Out [24]	

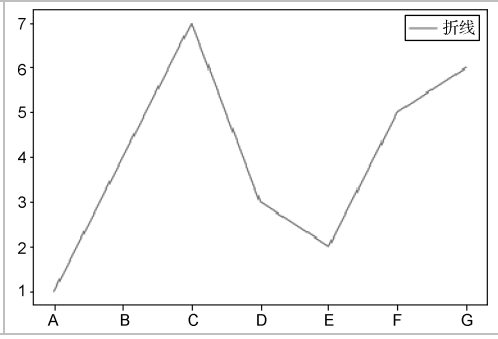
④ 文字函数。

text(x, y, labels,…)：在(x,y)处添加用 labels 指定的文字。

In [25]	plt.plot(X,Y);plt.text(2, 7, ' peak point')
Out [25]	

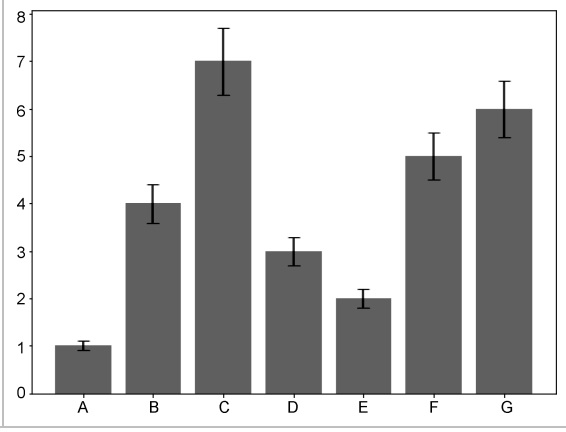
⑤ 图例。

绘制图形后，可使用 legend()函数给图形加图例，见下面的分析。

In [26]	plt.plot(X,Y,label=u'折线');plt.legend();
Out [26]	

(4) 误差条图

误差条图由带标记的线条组成，通常这些线条用于显示有关图中所显示的数据的标准差信息。

In [27]	s=[0.1,0.4,0.7,0.3,0.2,0.5,0.6] plt.bar(X,Y,yerr=s,error_kw={'capsize':5})
Out [27]	

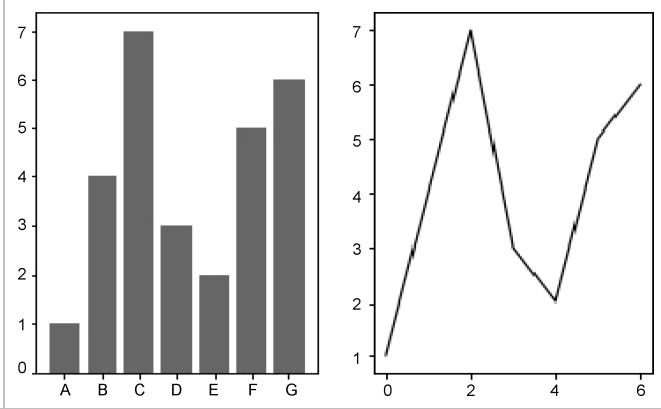
(5) 多图

在 matplotlib 下，一个 Figure 对象可以包含多个子图(Axes)，可以使用函数 subplot() 快速绘制，其调用形式如下：

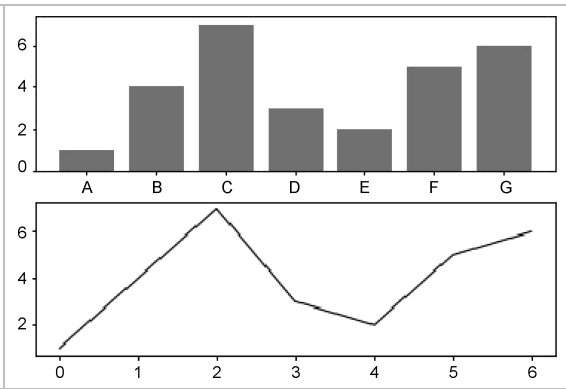
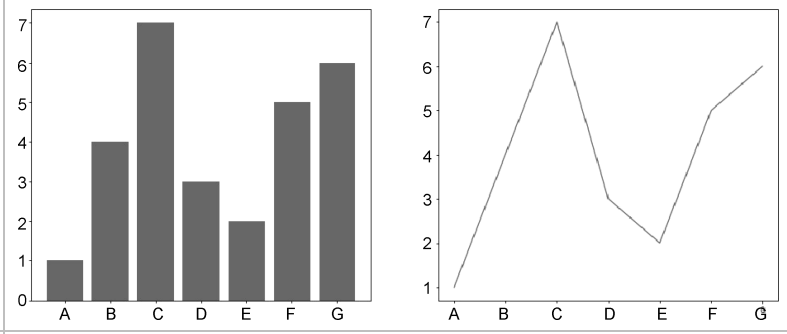
```
subplot(numRows, numCols, plotNum)
```

图表的整个绘图区域被分成 numRows 行和 numCols 列。然后按照从左到右、从上到下的顺序对每个子区域进行编号，左上子区域的编号为 1，plotNum 参数指定创建的 Axes 对象所在的区域。

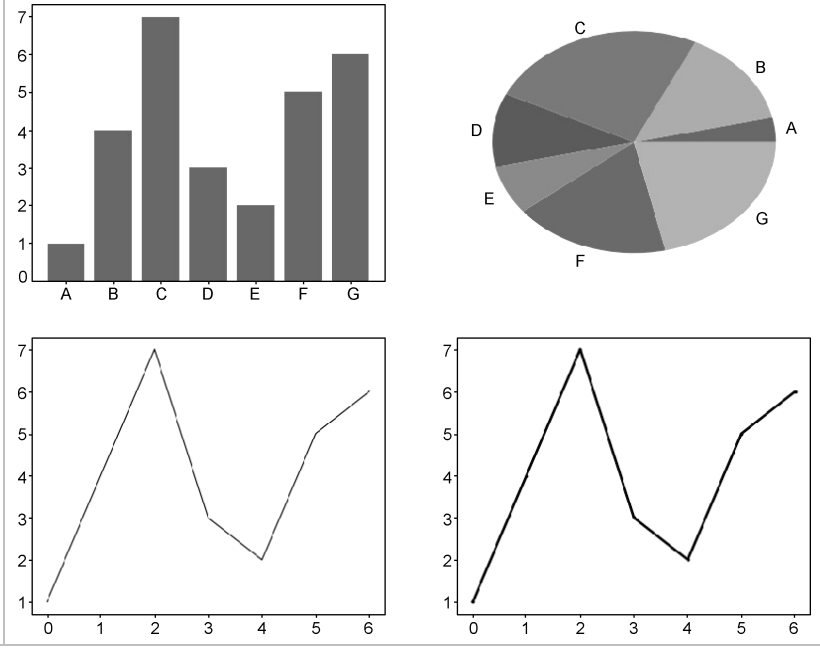
In [28]	'''一行绘制两个图形'''
	plt.subplot(121); plt.bar(X,Y);
	plt.subplot(122); plt.plot(Y);

Out [28]	

In [29]	'''一列绘制两个图形 '''
	plt.subplot(211); plt.bar(X,Y);
	plt.subplot(212); plt.plot(Y);

Out [29]	
In [30]	fig,ax = plt.subplots(1,2,figsize=(15,6)) #一页绘制两个图形 ax[0].bar(X,Y); ax[1].plot(X,Y);
Out [30]	

In [31]	fig,ax=plt.subplots(2,2,figsize=(15,12)) #一页绘制四个图形
	ax[0,0].bar(X,Y)
	ax[0,1].pie(Y,labels=X)
	ax[1,0].plot(Y);
	ax[1,1].plot(Y,'.-',linewidth=3);

Out [31]	

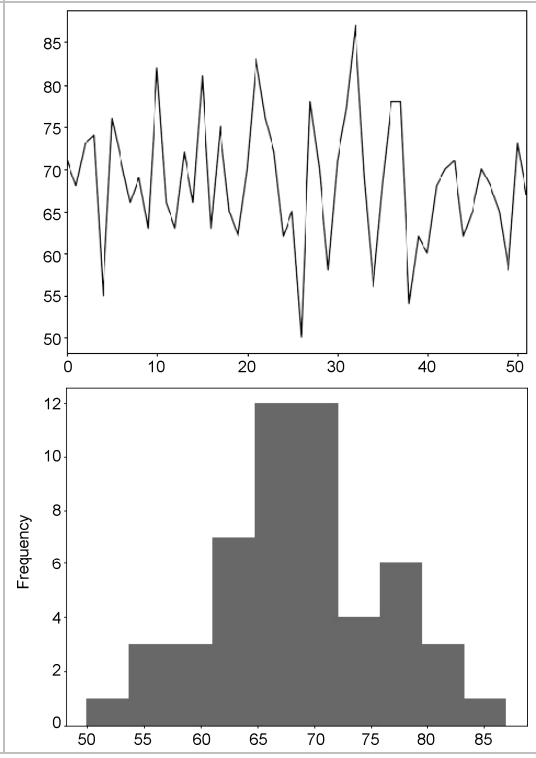
2.1.2.2 基于 pandas 的绘图

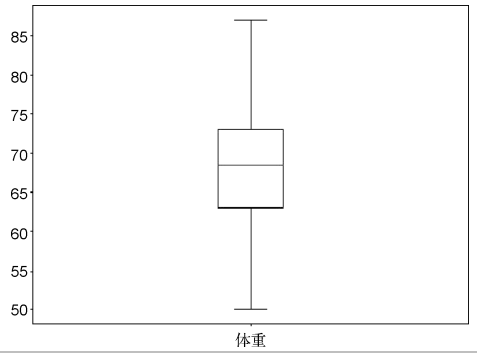
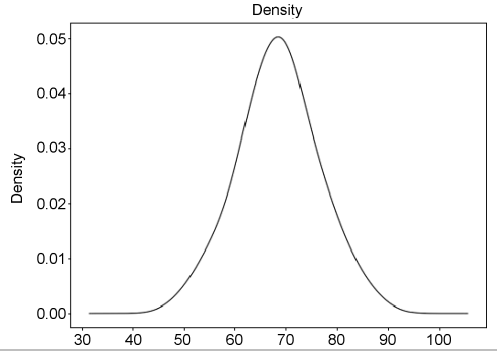
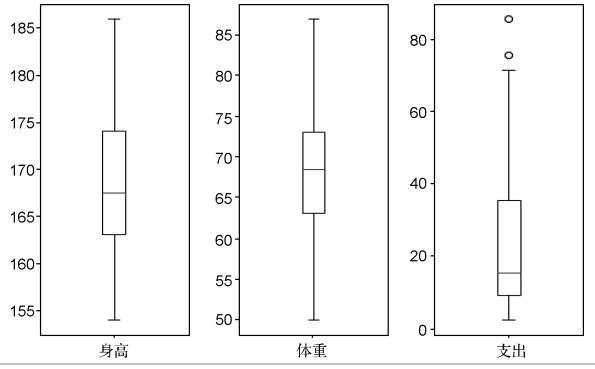
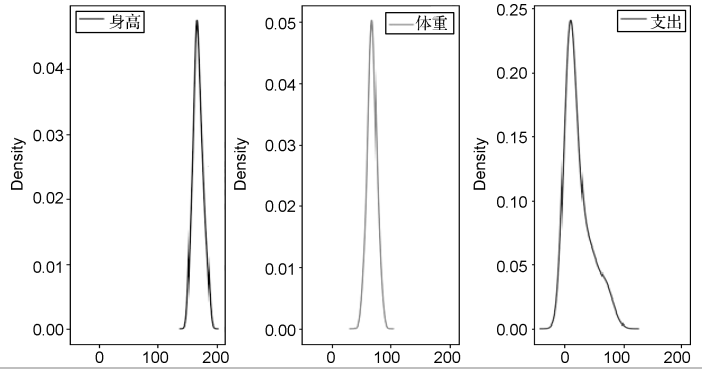
在 pandas 中，数据框有行标签、列标签及分组信息等。要制作一张完整的图表，原本需要一长串 matplotlib 代码，现在只需一两条简洁的语句。pandas 有许多能够利用 DataFrame 对象数据组织特点来创建标准图标的高级绘图方法(这些函数的数量还在不断增加)。

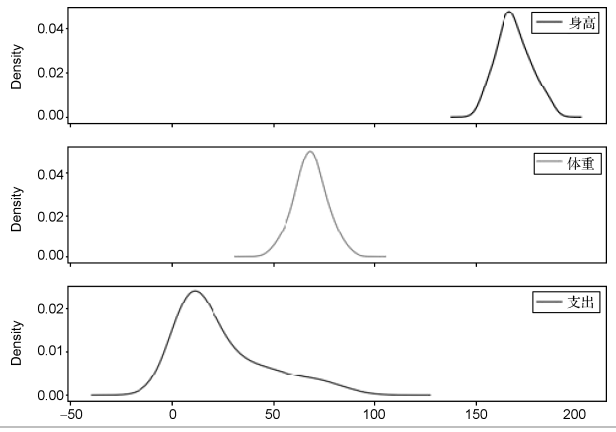
数据框 DataFrame 绘图时将每列作为一条图线绘制到一张图中，并用不同的线条颜色及不同的图例标签来表示，其基本格式如下。

```
DataFrame.plot(kind='line')
kind : 图类型
    'line' : (default)  #线图
    'bar':              #垂直条图
    'barh' :            #水平条图
    'hist' :            #直方图
    'box' :             #箱线图
    'kde' :             #核密度估计图，对柱状图添加概率密度线，同 'density'
    'area' :            #面积图
    'pie' :             #饼图
    'scatter' :         #散点图
```

(1)计量数据的 pandas 绘图

In [32]	BSdata['体重'].plot(kind='line'); BSdata['体重'].plot(kind='hist'); BSdata['体重'].plot(kind='box');
Out [32]	

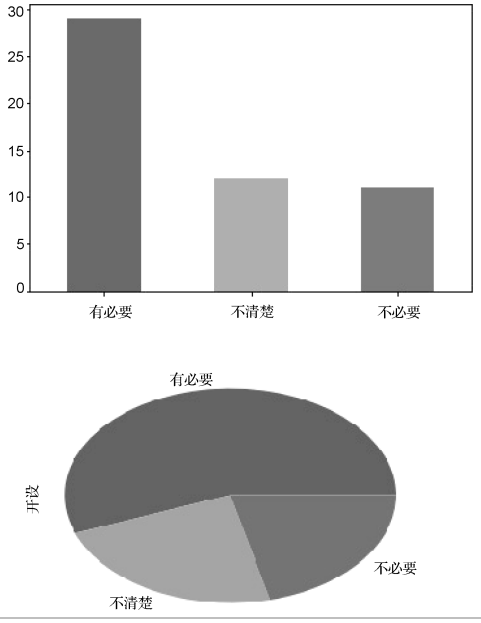
	
In [33]	BSdata['体重'].plot(kind='density',title='Density');
Out [33]	
In [34]	BSdata[['身高','体重','支出']].plot(subplots=True,layout=(1,3),kind='box')
Out [34]	
In [35]	BSdata[['身高','体重','支出']].plot(subplots=True,layout=(1,3),kind='density');
Out [35]	
In [36]	BSdata[['身高','体重','支出']].plot(subplots=True,layout=(3,1),kind='density');

Out [36]	

(2) 计数数据的 pandas 绘图

In [37]	T1=BSdata['开设'].value_counts();T1 pd.DataFrame({'频数':T1,'频率':T1/T1.sum()*100})
Out [37]	开设　频数　频率 有必要　29　55.769231 不清楚　12　23.076923 不必要　11　21.153846

In [38]	T1.plot(kind='bar'); #T1.sort_values().plot(kind='bar');
	T1.plot(kind='pie');

Out [38]	

2.2　数据的透视分析

数据透视分析通常是以透视表的形式进行的。透视表是一种交互式的表，可以进行某些计算，如求和与计数等。数据透视表可以动态地改变变量的布置，以便按照不同方式分析数据。

2.2.1 一维频数分析

频数分析，又称“次数分析”，数据的统计整理方式之一。通常按照某种标志（计数或计量）将数据分成若干组，分别统计各组数据的频数（有时包括频率），以反映数据分布在各组的情况。

一维频数分析即单变量数据的透视表分析。

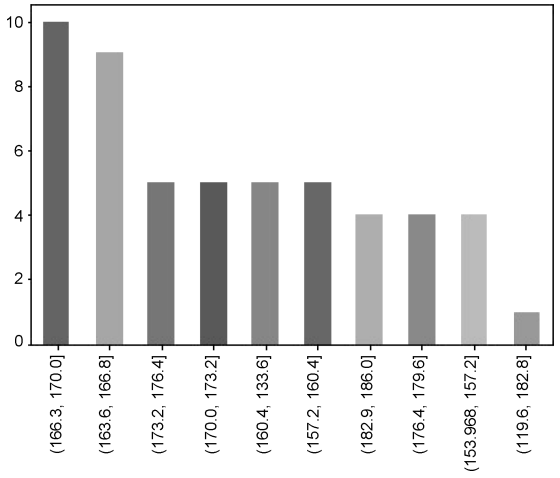
2.2.1.1 计数数据的频数分析

下面是课程开设数据的频数表与条图。

In [39]	BSdata['开设'].value_counts()
Out [39]	有必要 29 不清楚 12 不必要 11

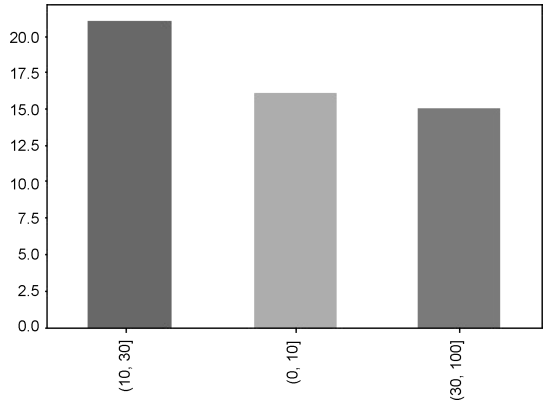
2.2.1.2 计量数据的频数分析

(1)身高数据的频数表与条图

In [40]	pd.cut(BSdata.身高,bins=10).value_counts()
Out [40]	(166.8, 170.0] 10 (163.6, 166.8] 9 (173.2, 176.4] 5 (170.0, 173.2] 5 (160.4, 163.6] 5 (157.2, 160.4] 5 (182.8, 186.0] 4 (176.4, 179.6] 4 (153.968, 157.2] 4 (179.6, 182.8] 1
In [41]	pd.cut(BSdata.身高,bins=10).value_counts().plot(kind='bar');
Out [41]	

大家可尝试制作 bins=[150,160,170,180,190,200]的分组表和条图。

(2)支出数据的频数表与条图

In [42]	pd.cut(BSdata.支出,bins=[0,10,30,100]).value_counts()
Out [42]	(10, 30] 21 (0, 10] 16 (30, 100] 15
In [43]	pd.cut(BSdata.支出,bins=[0,10,30,100]).value_counts().plot(kind='bar');
Out [43]	

2.2.2 二维集聚分析

2.2.2.1 计数数据的列联表

(1)二维列联表

Python 的 crosstab()函数可以把双变量分类数据整理成二维表形式。

In [44]	pd.crosstab(BSdata.开设,BSdata.课程)					
Out [44]	课程	概率统计	统计方法	编程技术	都学习过	都未学过
	开设					
	不必要	3	2	1	1	4
	不清楚	3	3	3	2	1
	有必要	5	10	6	7	1

行和列的合计可使用参数 margins=True。

In [45]	pd.crosstab(BSdata.开设,BSdata.课程,margins=True)						
Out [45]	课程	概率统计	统计方法	编程技术	都学习过	都未学过	All
	开设						
	不必要	3	2	1	1	4	11
	不清楚	3	3	3	2	1	12
	有必要	5	10	6	7	1	29
	All	11	15	10	10	6	52

对于二维表，我们经常要计算某个数据占行、列的比例或占总的比例，也就是边缘概率。用 normalize 参数，Python 可以简单地计算这些比例，normalize ='index'表示各数据占行的比例；normalize='columns'表示各数据占列的比例；normalize ='all'表示各数据占总和的比例。例如：

In [46]
```
pd.crosstab(BSdata.开设, BSdata.课程, margins=True, normalize='index')
```

Out [46]

课程	概率统计	统计方法	编程技术	都学习过	都未学过
开设					
不必要	0.2727	0.1818	0.0909	0.0909	0.3636
不清楚	0.2500	0.2500	0.2500	0.1667	0.0833
有必要	0.1724	0.3448	0.2069	0.2414	0.0345
All	0.2115	0.2885	0.1923	0.1923	0.1154

In [47]
```
pd.crosstab(BSdata.开设, BSdata.课程, margins=True, normalize='columns')
```

Out [47]

课程	概率统计	统计方法	编程技术	都学习过	都未学过	All
开设						
不必要	0.2727	0.1333	0.1	0.1	0.6667	0.2115
不清楚	0.2727	0.2000	0.3	0.2	0.1667	0.2308
有必要	0.4545	0.6667	0.6	0.7	0.1667	0.5577

In [48]
```
pd.crosstab(BSdata.开设, BSdata.课程, margins=True, normalize='all')
```

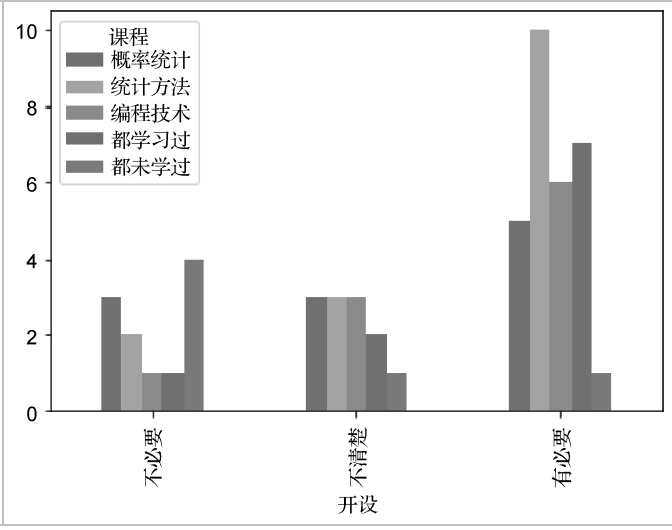
Out [48]

课程	概率统计	统计方法	编程技术	都学习过	都未学过	All
开设						
不必要	0.0577	0.0385	0.0192	0.0192	0.0769	0.2115
不清楚	0.0577	0.0577	0.0577	0.0385	0.0192	0.2308
有必要	0.0962	0.1923	0.1154	0.1346	0.0192	0.5577
All	0.2115	0.2885	0.1923	0.1923	0.1154	1.0000

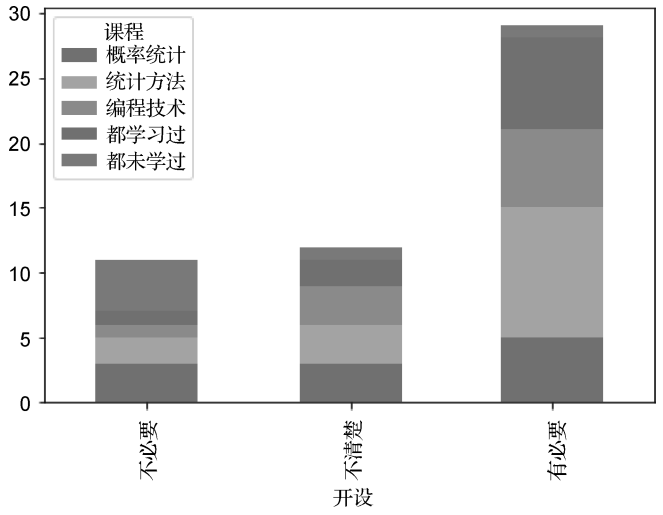
(2) 复式条图

In [49]
```
T2=pd.crosstab(BSdata.开设, BSdata.课程);T2
T2.plot(kind='bar');
```

Out [49]



条图用等宽直条的长短来表示相互独立的各指标数值大小，该指标可以是连续型变量的某汇总指标，也可以是分类变量的频数或构成比。各组直条间的间距应相等，其宽度一般与直条的宽度相等或为直条宽度的一半。Python 中作条图的函数是 bar()，在作条图前需要对数据进行分组。我们继续以上面的分类数据为例作条图，粗略分析变量的分布情况。

In [50]	T2.plot(kind='bar',stacked=True);
Out [50]	

其中 stacked 参数设置为 False 时，作出的图是分段式条图，为 True 时作出的图是并列式条图。

2.2.2.2 计量数据的集聚表

(1)分组 groupby()函数

pandas 提供了一项灵活高效的 groupby 功能，通过它可以以一种自然的方式对数据集进行切片、切块、摘要等操作；根据一个或多个键(可以是函数、数组或 DataFrame 列名)拆分 pandas 对象；计算分组摘要统计，如计数、平均值、标准差或用户自定义函数；对 DataFrame 的列应用各种函数。

① 按列分组。

注意：以下使用 groupby()函数生成一个中间分组变量，为 GroupBy 类型。

In [51]	BSdata.groupby(['性别']) type(BSdata.groupby(['性别']))
Out [51]	<pandas.core.groupby.DataFrameGroupBy object at0x000000000B748FD0> pandas.core.groupby.DataFrameGroupBy

② 按分组统计。

在分组结果的基础上应用 size()、sum()、count()等统计函数，可分别统计分组数量、不同列的分组和、不同列的分组数量。

In [52]	BSdata.groupby(['性别'])['身高'].mean()
Out [52]	性别 女　　165.360000 男　　171.444444 Name: 身高, dtype: float64
In [53]	BSdata.groupby(['性别'])['身高'].size()

Out [53]	性别 女　25 男　27
In [54]	BSdata.groupby(['性别','开设'])['身高'].mean()
Out [54]	性别　开设 女　不必要　165.1667 　　不清楚　164.5556 　　有必要　166.2000 男　不必要　180.2000 　　不清楚　173.3333 　　有必要　168.8421

③ 应用 agg()函数计算统计量。

对于分组的某一列或多列，应用 agg()可以对分组后的数据进一步计算，并可用于多个函数，如下面的均值和标准差函数 mean 与 std。

In [55]	BSdata.groupby(['性别'])['身高'].agg([np.mean, np.std])
Out [55]	mean　std 性别 女　165.360000　5.179125 男　171.444444　9.103395

④ 应用 apply()函数计算统计量。

apply()函数不同于 agg()函数的地方在于：前者作用于数据框的各个列，后者仅作用于指定的列。

In [56]	BSdata.groupby(['性别'])['身高','体重'].apply(np.mean)
Out [56]	身高　体重 性别 女　165.360000　66.240000 男　171.444444　70.592593
In [57]	BSdata.groupby(['性别','开设'])['身高','体重'].apply(np.mean)
Out [57]	身高　体重 性别　开设 女　不必要　165.1667　67.3333 　　不清楚　164.5556　64.6667 　　有必要　166.2000　67.0000 男　不必要　180.2000　79.2000 　　不清楚　173.3333　72.6667 　　有必要　168.8421　68.0000

2.2.3 多维透视分析

2.2.3.1 计数数据的透视分析

对计数数据，前面介绍了用 value_counts()函数生成一维频数表，用 croostab()函数生成二维列联表，其实 pivot_table()函数可以生成任意维统计表。下面用 pandas 包的 pivot_table()函数生成各种统计表，可以达到 Excel 等电子表格的透视表功能，且更为灵活方便。

用 pivot_table()函数生成计数数据的统计表时，参数 index 和 values 都为分类变量，aggfunc 通常取长度函数 len。

In [58]	BSdata.pivot_table(index=['性别'],values=['学号'],aggfunc=len)
Out [58]	学号 性别 女 25 男 27
In [59]	BSdata.pivot_table(values=['学号'],index=['性别','开设'],aggfunc=len)
Out [59]	学号 性别 开设 女 不必要 6 不清楚 9 有必要 10 男 不必要 5 不清楚 3 有必要 19
In [60]	BSdata.pivot_table(values=['学号'],index=['开设'],columns=['性别'],aggfunc=len)
Out [60]	学号 性别 女 男 开设 不必要 6 5 不清楚 9 3 有必要 10 19

2.2.3.2 计量数据的透视分析

pivot_table()函数也可以生成计量数据的统计表，这时参数 index 为分类变量，values 为数值变量，aggfunc 为要计算的统计量函数，如均值和标准差函数 mean 和 std。

In [61]	BSdata.pivot_table(index=['性别'],values=["身高"],aggfunc=np.mean)
Out [61]	身高 性别 女 165.360000 男 171.444444
In [62]	BSdata.pivot_table(index=['性别'],values=["身高"],aggfunc=[np.mean,np.std])
Out [62]	mean std 身高 身高 性别 女 165.360000 5.179125 男 171.444444 9.103395
In [63]	BSdata.pivot_table(index=["性别"],values=["身高","体重"])
Out [63]	体重 身高 性别 女 66.240000 165.360000 男 70.592593 171.444444

2.2.3.3 复合数据的透视分析

pivot_table()函数还可以生成复合数据(计数与计量数据)的统计表，这时参数变量既

可以是分类变量，也可以是数值变量。统计量函数 aggfunc 可包含计数和计量函数，如长度、均值和标准差函数 len、mean 和 std。

In [64]
```
BSdata.pivot_table('学号',['性别','开设'],'课程',aggfunc=len,
                   margins=True,margins_name='合计')
```

Out [64]

性别	开设	概率统计	统计方法	编程技术	都学习过	都未学过	合计
女	不必要	1.0	1.0	NaN	1.0	3.0	6.0
	不清楚	2.0	1.0	3.0	2.0	1.0	9.0
	有必要	NaN	5.0	3.0	2.0	NaN	10.0
男	不必要	2.0	1.0	1.0	NaN	1.0	5.0
	不清楚	1.0	2.0	NaN	NaN	NaN	3.0
	有必要	5.0	5.0	3.0	5.0	1.0	19.0
合计		11.0	15.0	10.0	10.0	6.0	52.0

In [65]
```
BSdata.pivot_table(['身高','体重'],['性别',"开设"],aggfunc=[len,np.mean,np.std] )
```

Out [65]

		len		mean		std	
		体重	身高	体重	身高	体重	身高
性别	开设						
女	不必要	6	6	67.333333	165.166667	3.326660	2.786874
	不清楚	9	9	64.666667	164.555556	6.000000	6.502136
	有必要	10	10	67.000000	166.200000	5.333333	5.308274
男	不必要	5	5	79.200000	180.200000	4.438468	4.147288
	不清楚	3	3	72.666667	173.333333	5.033223	5.686241
	有必要	19	19	68.000000	168.842105	9.165151	9.124224

数据及练习 2

2.1 调查数据：某公司对财务部门人员的抽烟情况进行调查，结果为：否,否,否,是,是,否,否,是,否,是,否,否,是,是,否,是,否,是,是,否,是,否,否,是,是。

请用 value_count()函数统计人数，并绘制条图，按颜色区分男女。

2.2 医学数据：对一组 50 人的饮酒者所饮酒类进行调查，把饮酒者按红酒(1)、白酒(2)、黄酒(3)、啤酒(4)分成四类。调查数据如下：3,4,1,1,3,4,3,3,1,3,2,1,2,1,3,4,1,1,3,4,3,3,1,3,2,1,2,1,2,3,2,3,1,1,1,1,4,3,1,2,3,2,3,1,1,1,1,4,3,1。

(1) 请用 value_count()函数统计饮酒人数，用 pie()函数绘制饼图，并按颜色和文字区分酒的类型。

(2) 请用 value_count()函数构建自己的计数频数表函数。

(3) 请自定义一个计数数据的频数表生成函数和频数图绘制函数。

2.3 工资数据：某企业财务部员工的月工资数据如下：2050,2100,2200,2300,2350,2450,2500,2700,2900,2850,3500,3800,2600,3000,3300,3200,4000,3100,4200,3500。

(1) 试用 mean()、median()、var()、sd()函数求数据的均值、中位数、方差、标准差。

(2) 绘制该数据的散点图和直方图，应用 hist()函数构建自己的计量频数表函数。

(3) 请自定义一个计量数据的频数表生成函数和频数图绘制函数。

2.4 经理年薪。收集某沿海发达城市 66 个 2015 年年薪超过 10 万元的公司经理的收入(单位：万元)：11,19,14,22,14,28,13,81,12,43,11,16,31,16,23,42,22,26,17,22,13,27,108,16,43,82,14,11,51,76,28,66,29,14,14,65,37,16,37,35,39,27,14,17,13,38,28,40,85,32,25,26,16,120,54,40,18,27,16,14,33,29,77,50,19,34。

(1) 我们能对这些薪酬的分布状况做何分析？

(2) 试编写计算基本统计量的函数来分析数据的集中趋势和离散程度。

(3) 试分析为何该数据的均值和中位数差别如此之大，方差、标准差在此有何作用？如何正确分析该数据的集中趋势和离散程度？

(4) 绘制该数据的散点图和直方图。

(5) 请用自定义函数生成频数表和频数图。

2.5 economics 数据集(来自 R 语言的 ggplot2 包)给出了美国经济增长变化的数据。该数据是数据框格式的，共 478 行，6 个变量，变量如下。

date：日期，单位为月份；

psavert：个人存款率；

pce：个人消费支出，单位为十亿美元；

uemploy：失业人数，单位为千人；

unempmed：失业时间中位数，单位为周；

pop：人口数，单位为千人。

请用 matplotlib，seaborn 和 ggplot 三种绘图系统绘制：

(1) 以 date 为横坐标，unemploy/pop 为纵坐标画线图。

(2) 以 date 为横坐标，unempmed 为纵坐标画线图。

第 3 章　简单数据的统计分析

随机变量的概率分布对现实世界的建模和分析发挥着重要作用。有时，理论分布与收集到的某过程的历史数据十分贴近。有时，可以先对某过程的基本特性做先验性判断，然后不需要收集数据就可以选出合适的理论分布。在这两种情况下，均可用理论分布来回答现实中所遇到的问题，也可以从分布中生成一些随机数来模拟现实的行为。

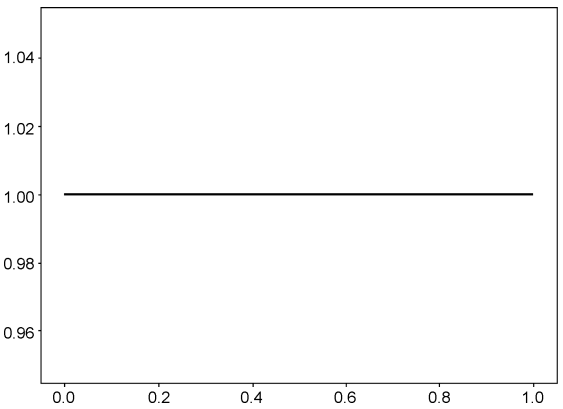
随机变量及其分布虽然不是进行数据处理的重点，但通过这些学习，我们可以进一步掌握 Python 的编程技巧，为下一步的统计分析和统计建模打下基础。

3.1　随机变量及其分布

3.1.1　均匀分布

这里“均匀”是指随机点落在区间(*a*,*b*)内任一点的机会是均等的，从而在相等的小区间上的概率相等，在任一区间(*a*,*b*)的，随机变量 *X* 的概率密度函数为一个常数。

$$y = P(x) = 1/(b-a) \quad (a<x<b)$$

In [1]	a=0;b=1;y=1/(b−a) plt.plot(a,y); plt.hlines(y,a,b);
Out [1]	

均匀分布是随机抽样和随机模拟的基础，可用 randint()和 uniform()函数产生均匀随机数。

(1)整数随机数

In [2]	import random random.randint(10,20)　　#[10,20]内的随机整数
Out [2]	19

(2) 实数随机数

In [3]	random.uniform(0,1)　　#[0,1)内的随机实数
Out [3]	0.5199571827163924

(3) 整数随机数列

In [4]	import numpy as np np.random.randint(10,21,9)　　#[10,20]内的 9 个随机整数数组
Out [4]	array([19, 18, 15, 19, 18, 12, 11, 14, 14])

(4) 实数随机数列

In [5]	np.random.uniform(0,1,10)　　#[0,1]内的 10 个随机数=np.random.rand(10)
Out [5]	array([0.84817348,0.973288,0.15355783,0.00825514,0.95753485, 0.18075714,0.50495058,0.49185524,0.83905195,0.01591327])

3.1.2　正态分布

(1) 正态分布函数

正态分布是统计分析的最主要分布。正态分布是古典统计学的核心，它有两个参数：位置参数均值μ，尺度参数标准差σ。正态分布的图形如倒立的钟，且分布对称。现实生活中，很多变量是服从正态分布的，比如人的身高、体重和智商 IQ。

① 密度函数：正态分布的概率密度函数有如下形式。

$$P(x)=\frac{1}{\sqrt{2\pi}\sigma}\mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

它的图形是对称的钟形曲线，常称为正态曲线。

② 分布函数：正态分布含有两个参数 μ 和 σ，记为 $x\sim N(\mu,\sigma^2)$。

③ 均值：$E(x)=\mu$。

④ 方差：$\mathrm{Var}(x)=\sigma^2$。

⑤ 标准差：σ。

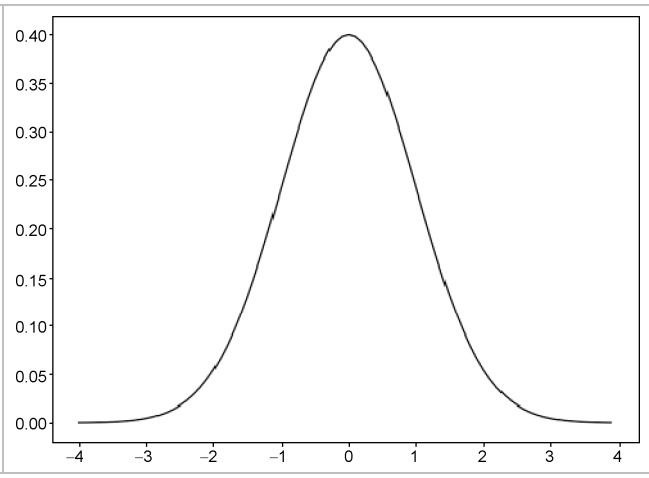
(2) 标准正态分布

可用正态化变换 $z=(x-\mu)/\sigma$ 将一般正态分布 $x\sim N(\mu,\sigma^2)$ 转换为标准正态分布 $z\sim N(0,1)$。

标准正态分布概率密度函数为 $P(z)=\frac{1}{\sqrt{2\pi}}\mathrm{e}^{-\frac{z^2}{2}}$。

① 标准正态分布曲线。

In [6]	from math import sqrt,pi
	x=np.linspace(-4,4,50);
	y=1/sqrt(2*pi)*np.exp(-x**2/2);
	plt.plot(x,y);

Out [6]	

② 标准正态分位数。

标准正态分布的α分位数是这样一个数，它的左侧面积恰好为α，它的右侧面积恰好为 $1-\alpha$，分位数 z_α是满足下列等式的实数：

$$P(z \leqslant z_\alpha) = \alpha, \text{ 且 } z_{0.5} = 0, \quad z_\alpha = -z_{1-\alpha}$$

求标准正态分布 $P(z \leqslant 2)$的累积概率。

In [7]	import scipy.stats as st P=st.norm.cdf(2);P	#加载统计方法包
Out [7]	0.9772498680518208	

已知标准正态分布累积概率为 $P(|z| \leqslant a)=0.95$，求对应的分位数 a。

In [8]	za=st.norm.ppf(0.95);za	#单侧
Out [8]	−1.9599639845400545	
In [9]	[st.norm.ppf(0.025),st.norm.ppf(0.975)]	#双侧
Out [9]	[−1.9599639845400545, 1.959963984540054]	

(3) 正态随机数

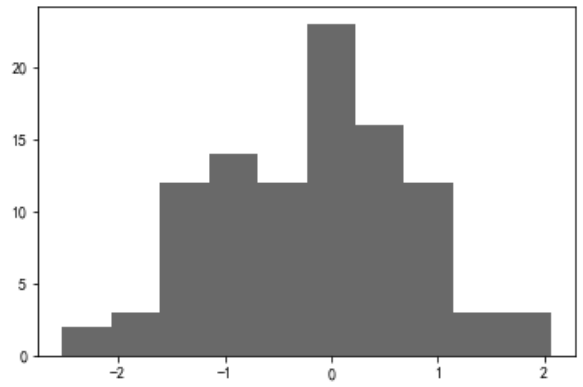
正态分布随机数的生成函数是 random.normal(mean=0,sd=1,n)，其中，n 表示生成的随机数数量(或正态随机样本数)，mean 是正态分布的均值，sd 是正态分布的标准差。

① 标准正态随机数。

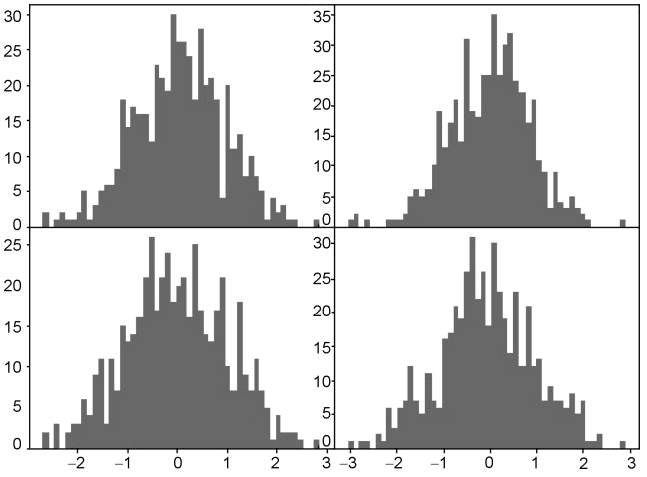
In [10]	np.random.normal(0,1,5)	#生成 5 个标准正态分布随机数
Out [10]	array([-1.26460061,-0.31995156,1.50601752,-0.53570903,.13590464])	

随机产生 1000 个标准正态分布随机数，作其概率直方图，然后再添加正态分布的密度函数线。

In [11]	z=np.random.normal(0,1,100) plt.hist(z)

Out [11]	

In [12]	'''一页绘制四个正态随机图 '''
	fig,ax = plt.subplots(2,2)
	for i in range(2):
	for j in range(2):
	ax[i,j].hist(np.random.normal(0,1,500),bins = 50);
	plt.subplots_adjust(wspace = 0,hspace=0);

Out [12]	

② 一般正态随机数。

In [13]	np.random.normal(10,4,5) #产生 5 个均值为 10、标准差为 4 的正态随机数
Out [13]	array([9.4957839,8.50000695,13.34632939,14.815103,8.7812742])

3.2 随机模拟及其应用

3.2.1 随机模拟方法

随机模拟，也称蒙特卡罗(Monte Carlo)模拟，是以概率统计的理论为基础的一种模拟方法，蒙特卡罗模拟又称统计实验法。蒙特卡罗模拟将所求解的问题与某个概率模型联系在一起，并在计算机上随机模拟，以获得问题的近似解。

蒙特卡罗是摩纳哥国的著名赌城，第二次世界大战期间，冯·诺依曼和乌拉姆秘密研制原子弹，并将蒙特卡罗作为秘密代号，对裂变物质的中子随机扩散进行模拟。

蒙特卡罗模拟的最突出特点是，模型的解是试验生成的，而不是计算出来的。它的主要优点可以归纳为如下三点：

① 蒙特卡罗模拟方法和程序结构比较简单。蒙特卡罗模拟只需要对总体进行大量的重复抽样，然后再求取这些模拟结果的期望值，期望值就是最终结果。蒙特卡罗模拟便于理解、使用和推广，适用范围非常广泛。

② 收敛速度与问题维数无关。蒙特卡罗模拟的收敛是概率意义下的收敛，无论问题维数多大，它的收敛速度都是一样的。所以，在低维情况下，它的速度看起来比较慢，但在高维情况下，就比其他数值计算方法的速度快得多。

③ 蒙特卡罗模拟方法的适用性非常强。蒙特卡罗模拟在解决问题时受问题条件的限制较小，而且不需要太多前提假设，和模拟对象的实际情况较为接近。而其他数值方法受问题条件限制比较大，适用性不强。

如果知道了某个概率分布，就可以通过 Python 模拟生成服从该分布的随机变量。随机数的生成是在进行统计模拟时进行随机抽样的基础。随机数最早是手工产生的，现在则由计算机生成。例如，金融计算的模拟常常涉及金融产品价格或收益率的分布，很多时候需要模拟价格或者收益率的变动过程。

3.2.2 模拟大数定律

设随机事件 E 的样本空间中只有有限个样本点，即$\Omega=\{\omega_1,\omega_2,\cdots,\omega_n\}$，其中 n 为样本点总数，每个样本点$\omega_t(t=1, 2, \cdots, n)$的出现是等可能的，并且每次试验有且仅有一个样本点发生，则称这类现象为古典概型。若事件 A 包含 m 个样本点，则事件 A 的概率定义为

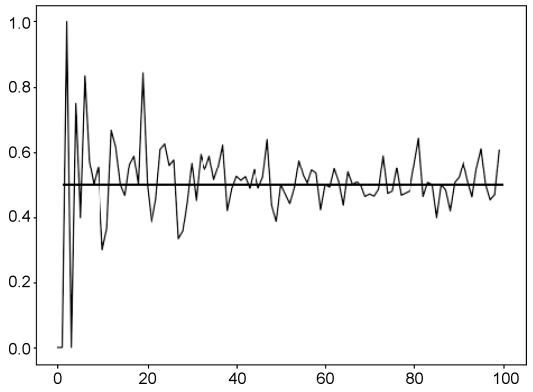
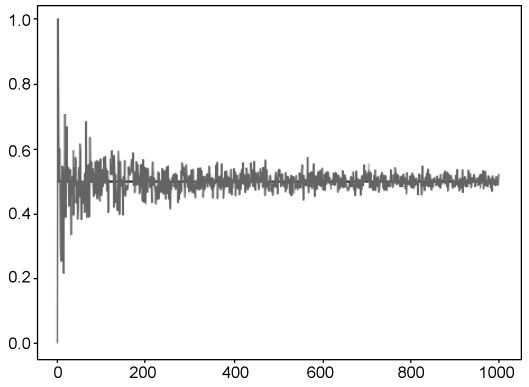
$$P(A)=\frac{m}{n}=\frac{\text{事件}A\text{包含的基本事件数}}{\text{基本事件总数}}$$

Bernoulli 大数定律：设 n_A 是 n 次独立重复试验中事件 A 发生的次数，P 是事件 A 在每次试验中发生的概率，则对于任意正数ε，有

$$\lim_{n\to\infty}P\left\{\left|\frac{n_A}{n}-p\right|<\varepsilon\right\}=1$$

Bernoulli 大数定律揭示了“频率稳定于概率”说法的实质，下面通过扔硬币的方式模拟大数定律。

In [14]	def Bernoulli(N=100):
	    p=np.zeros(N)
	    for n in range(1,N):
	        f=np.random.randint(0,2,n) #[0,1]
	        m=sum(f)
	        p[n]=m/n

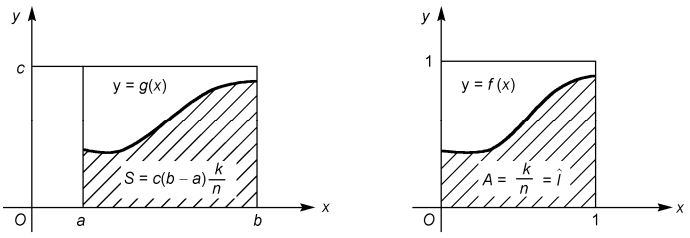
	plt.plot(p);plt.hlines(0.5,1,N) Bernoulli()
Out [14]	
In [15]	Bernoulli(1000)
Out [15]	

3.2.3 模拟方法求积分

随机数的最早应用之一是积分的计算，下面给出用模拟技术求定积分的方法。

$$I=\int_a^b g(x)\mathrm{d}x$$

解：下面图(a)的阴影面积表示定积分 I 的值。为简化问题，将函数限制在单位正方形($0\leqslant x\leqslant 1$，$0\leqslant y\leqslant 1$)内，如图(b)所示。只要函数 $g(x)$ 在区间$[a,b]$内有界，就可以适当选择坐标轴的比例尺度，得到下图的形式。



(a) (b)

用蒙特卡罗方法求积分的示意图

首先考虑图(b)的情况，计算定积分

$$I=\int_0^1 f(x)\mathrm{d}x$$

令 x，y 为相互独立的(0，1)区间上的均匀随机数，在单位正方形内随机投掷 n 个点 (x_i, y_i) (i=1,2,⋯,n)，若第 i 个随机点 (x_i,y_i) 落于曲线 $f(x)$ 下的区域内(图(b)内有阴影的区域)，表明第 i 次试验成功，这相应于满足概率模型 $y_i \leqslant f(x_i)$。

设成功的总点数有 k 个，总的试验次数为 n，则由强大数定律，有 $\lim\limits_{n\to\infty}\dfrac{k}{n}=P$，从而有 $\hat{I}=\dfrac{k}{n}\approx P$。

显然，概率 P 即图(b)的阴影部分面积 A，从而，随机点落在该阴影部分的概率 P 恰是所求积分的估计值 $\hat{I}$。

用模拟方法求定积分的一般方法：

如要计算 $I=\int_a^b g(x)\mathrm{d}x$，令 $y=(x-a)/(b-a)$，则有

$$\begin{cases}\mathrm{d}y=\mathrm{d}x/(b-a)\\ I=\int_a^b g(x)\mathrm{d}x=\int_0^1 g(a+(b-a)y)(b-a)\mathrm{d}y=\int_0^1 h(y)\mathrm{d}y\end{cases}$$

式中，$h(y)=(b-a)g(a+(b-a)y)$，y 是[0,1]区间上均匀分布的随机数。

例如，要计算标准正态分布曲线下的概率 $\int_{-1}^{1}\dfrac{1}{\sqrt{2\pi}}\mathrm{e}^{-\frac{x^2}{2}}\mathrm{d}x$，由于被积函数不可积，所以要用蒙特卡罗模拟进行数值计算。

In [16]	from math import sqrt,pi,exp
	def g(x):
	　　return (1/sqrt(2*pi))*exp(-x**2/2)
	def I(n,a,b,g):
	　　x=np.random.uniform(0,1,n)
	　　return sum([(b–a)*g(a+(b–a)*y) for y in x])/n
	I(10000,–1,1,g)

Out [16]	0.6823545802138847

下面用 Python 的科学技术包 scipy 中的求积分函数直接求上述积分。

In [17]	from scipy.integrate import quad quad(g,–1,1)
Out [17]	(0.682689492137086, 7.579375928402476e–15)

3.3 单变量统计分析模型

相关分析是指通过对大量数字资料的观察，消除偶然因素的影响，探求现象之间相关关系的密切程度和表现形式。研究现象之间相关关系的理论方法称为相关分析法。

在经济管理中，各经济变量常常存在密切的关系，如经济增长与财政收入、人均收入与消费支出等。这些关系大都是非确定的关系，一个变量发生变动会影响其他变量，使其发生变化，其变化具有随机的特性，但是仍然遵循一定的规律。

回归分析研究两变量之间的依存关系，将变量区分出自变量和因变量，并研究确定自变量和因变量之间具体关系的方程形式。分析中所形成的自变量和因变量之间的关系式称为回归模型，其中以一条直线方程表明两变量依存关系的模型叫一元线性回归模型(也称直线回归模型)。回归分析的主要步骤包括建立回归模型、求解回归模型中的参数、对回归模型进行检验等。

3.3.1 单变量线性相关模型

在所有相关分析中，最简单的是两个变量之间的一元线性相关(也称简单线性相关)，它只涉及两个变量。而且一个变量数值发生变动，另一个变量的数值随之发生大致均等的变动，从平面图上观察，其各点的分布近似地表现为一条直线，这种相关关系称为线性相关。

线性相关分析是用相关系数来表示两个变量间相互的线性关系，并判断其密切程度的统计方法。总体相关系数通常用 ρ 表示，其计算公式为

$$\rho=\frac{\mathrm{Cov}(x,y)}{\sqrt{\mathrm{Var}(x)\cdot\mathrm{Var}(y)}}=\frac{\sigma_{xy}}{\sigma_x^2\sigma_y^2}$$

式中，σ_x^2 为变量 x 的总体方差，σ_y^2 为变量 y 的总体方差，σ_{xy} 为变量 x 与变量 y 的总体协方差。相关系数 ρ 没有单位，在–1～+1 范围内波动，其绝对值愈接近 1，两个变量间的直线相关愈密切；愈接近 0，相关愈不密切。

3.3.1.1 相关系数的计算

在实践中，通常要计算样本的线性相关系数，计算公式为

$$r=\frac{s_{xy}}{\sqrt{s_x^2\cdot s_y^2}}=\frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2\sum(y-\bar{y})}}$$

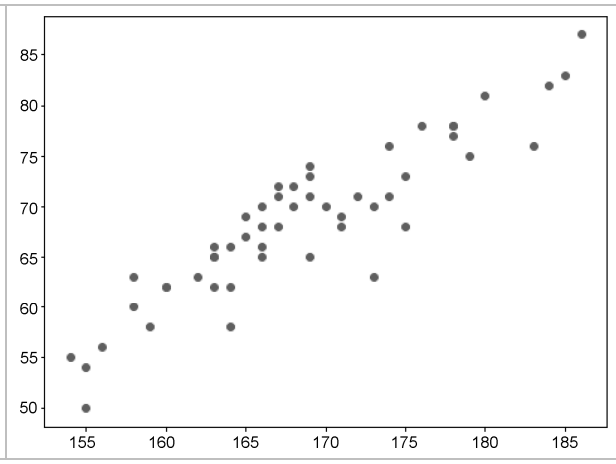
式中，s_x^2 为变量 x 的样本方差；s_y^2 为变量 y 的样本方差；s_{xy} 为变量 x 与变量 y 的样本协方差。

Person 相关系数 r 的取值范围是[–1,1]。$-1<r<0$ 表示具有负线性相关，越接近–1，负相关性越强；$0<r<1$ 表示具有正线性相关，越接近 1，正相关性越强；$r=-1$ 表示具有完全负线性相关；$r=1$ 表示具有完全正线性相关；$r=0$ 表示两个变量不具有线性相关性。相关系数是协方差的标准化形式，它消除了单位的影响。

在研究生的开课信息调查数据中，考察一下学生身高和体重的相关关系，下页图是身高和体重的散点图。

(1)散点图

In [18]	x=BSdata.身高;y=BSdata.体重 plt.plot(x, y,'o');　　#plt.scatter(x,y);

Out [18]	

(2) 协方差及相关系数

Python 中自带的计算协方差和相关系数的函数是.cov() 和.corr() 。

In [19]	x.cov(y)	#y.cov(x)
Out [19]	56.382352941176478	
In [20]	x.corr(y)	#y.corr(x)
Out [20]	0.9118170987010521	

这里相关系数为正值, 并且较大(>0.9)，说明学生身高与体重间呈现较强的线性相关性。至于相关系数是否显著，尚需进行假设检验。

3.3.1.2 相关系数的检验

与其他统计量一样，样本相关系数 r 也有抽样误差。从同一总体内抽取若干大小相同的样本,各样本的相关系数总有波动。要判断不等于 0 的 r 值是来自总体相关系数 $\rho=0$ 的总体，还是来自 $\rho\neq0$ 的总体，必须进行显著性检验，Python 的 Pearson 相关系数的检验函数为 st.pearsonr。

由于来自 $\rho=0$ 的总体的所有样本相关系数呈对称分布，故 r 的显著性可用 t 检验来进行。对 r 进行 t 检验的步骤如下。

① 建立检验假设：$H_0:\rho=0$，$H_1:\rho\neq0$，$\alpha=0.05$。

② 计算相关系数 r 的 t 值：

$$t_r=\frac{r-\rho}{s_r}=\frac{r}{\sqrt{(1-r^2)/(n-2)}}$$

(3) 计算 P 值，给出结论。

In [21]	st.pearsonr(x,y)	#Pearson 相关及检验
Out [21]	(0.9118170987010525, 5.74732931644513e–21)	#(系数，P 值)

由于 P=5.747e–21<0.05，于是在 $\alpha=0.05$ 置信水平上拒绝 H_0，接受 H_1，可以认为学生身高与体重间具有显著的线性相关性。

3.3.2 单变量线性回归模型

3.3.2.1 单变量线性回归模型估计

在因变量和自变量的散点图中，如果趋势大致呈直线形，即

$$y = \beta_0 + \beta_1 x + e$$

则可拟合一条直线方程，这里 e 为误差(error)，相应的直线回归模型为

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x = a + bx$$

式中，$\hat{y}$ 表示因变量 y 的估计值。x 为自变量的实际值。a，b 为待估参数，其几何意义：a 是直线方程的截距，为常数项，b 是斜率，称为回归系数；其经济意义：a 是当 x 为零时 y 的估计值，b 是当 x 每增加一个单位时 y 增加的数量。

拟合回归直线的目的是找到一条理想的直线，用直线上的点来代表所有相关点。数理统计证明，用最小平方法拟合的直线最理想，最具有代表性。计算 a 与 b 常用普通最小二乘法(OLS)。

由身高(x)和体重(y)的散点图可见，虽然 x 与 y 间有直线趋势存在，但并不是一一对应的，每个值 x_i 与 y_i (i=1,2,⋯,n)用回归方程估计的值 $\hat{y}_i$ (也称拟合值，即直线上的点)或多或少存在一定的差距，这些差距可以用 $\hat{e} = y - \hat{y}$ 来表示，称为估计误差或残差(resid)。要使回归方程比较“理想”，就应该使估计误差尽量小，也就是使估计误差平方和

$$Q = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} [y_i - (a + bx_i)]^2$$

达到最小。对 Q 求关于 a 和 b 的偏导数，并令其等于零，可得

$$\begin{cases} b = \dfrac{\sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n} (x_i - \overline{x})^2} \\ a = \overline{y} - b\overline{x} \end{cases}$$

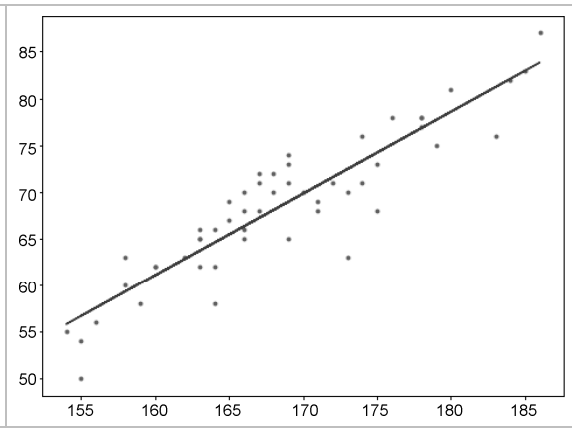
下面是 Python 的 OLS 估计方法。

In [22]	import statsmodels.api as sm fm1=sm.OLS(y,sm.add_constant(x)).fit() fm1.params	#加载线性回归模型包 #普通最小二乘，加常数项 #参数估计值
Out [22]	const −79.282827 #a 身高 0.876949 #b	

回归直线拟合图如下。

In [23]	yfit=fm1.fittedvalues plt.plot(x, y,'.',x,yfit, 'r-');

Out [23]



由散点图观察实测样本资料是否存在一定的协同变化趋势，这种趋势是否是线性的，根据是否有线性趋势确定应拟合直线还是曲线。由本例资料绘制的散点图可见，身高与体重之间存在明显的线性趋势，所以可考虑建立直线回归方程。

Python 作为一种面向对象语言，与其他数据分析软件相比，最大的优势就是输出结果简洁，而把大量的统计结果作为对象保存起来，以供后期使用。比如，前面的 fm1 就是一个线性回归模型的对象，其中包含进一步分析的统计量，如前面的参数估计值(params)、拟合值(fittedvalues)等。

3.3.2.2 单变量线性回归模型检验

由样本资料建立回归方程的目的是对两变量的回归关系进行统计推断，也就是对总体回归方程进行参数估计和假设检验。前面我们对回归模型的系数进行了估计，下面对回归系数进行假设检验。

由于抽样误差的存在，样本回归系数往往不会恰好等于总体回归系数。如果总体回归系数为 0，那么模型就是一个常数，无论自变量如何变化，都不会影响因变量，回归方程就没有意义。由样本资料计算得到的样本回归系数不一定为 0，所以有必要对估计得到的样本回归系数进行检验。

(1)常数项β_0的假设检验

原假设 H_0:β_0=0，以判断直线是否通过原点。检验统计量为

$$t_{\hat{\beta}_0}=\frac{\hat{\beta}_0-\beta_0}{s_{\hat{\beta}_0}}\sim t(n-2)$$

式中，分母为常数项的标准误。

(2)回归系数β_1的假设检验

原假设 H_0:β_1=0，直线方程不存在。检验统计量为

$$t_{\hat{\beta}_1}=\frac{\hat{\beta}_1-\beta_1}{s_{\hat{\beta}_1}}\sim t(n-2)$$

式中，分母为样本回归系数的标准误。

下面对前面建立的回归模型进行假设检验：

In [24]	fm1.tvalues　　#系数 t 的检验值
Out [24]	Intercept　−8.414938 身高　15.702810
In [25]	fm1.pvalues　　#系数 t 的检验概率
Out [25]	Intercept　3.820690e−11 身高　5.747329e−21
In [26]	pd.DataFrame({'b 估计值':fm1.params,'t 值':fm1.tvalues,'概率 p':fm1.pvalues})
Out [26]	b 估计值　t 值　概率 p const　−79.282827　−8.414938　3.820690e−11 身高　0.876949　15.702810　5.747329e−21

由于回归系数的 P = 5.747e−21< 0.05，于是在 $\alpha = 0.05$ 水平处拒绝原假设 H_0，接受备择假设 H_1，认为回归系数有统计学意义，变量间存在回归关系。

通常，我们更喜欢用公式的方式来建立线性回归模型，并用回归系数检验表来显示。

In [27]	import statsmodels.formula.api as smf　　#根据公式建立回归模型 fm2=smf.ols('体重～身高', BSdata).fit() fm2.summary2().tables[1]　　#回归系数检验表
Out [27]	Coef.　Std.Err.　t　P>|t|　[0.025　0.975] Intercept　−79.282827　9.421677　−8.414938　3.820690e−11　−98.206823　−60.358831 身高　0.876949　0.055847　15.702810　5.747329e−21　0.764778　0.989121

3.3.2.3　单变量线性回归模型预测

建立模型有三个主要作用：①进行影响因素分析；②进行估计；③用来预测。前面主要探讨了线性回归模型的因素分析，下面分别用模型进行估计和预测。估计指在自变量范围内对因变量的估算，预测指在自变量范围外对因变量的推算。Python 所用的函数都是 predict()（相当于将自变量值代入模型中计算），下面是身高与体重模型的估计与预测。

In [28]	fm2.predict(pd.DataFrame({'身高': [178,188,190]}))　　#估计与预测
Out [28]	0　76.814154 1　85.583648 2　87.337546

数据及练习 3

3.1　cars 数据集（来自 R 语言的 datasets 包）给出了 1920 年记录的汽车行驶速度 speed 和刹车距离 dist 的数据。

(1) 作 speed 与 dist 的散点图，并以此判断 speed 与 dist 之间是否大致呈线性关系。

(2) 计算 speed 与 dist 的相关系数并进行假设检验。

(3) 建立 speed 对 dist 的 OLS 回归模型，并给出常用统计量。

(4) 预测当 speed=30 时，dist 等于多少。

3.2 freeny 数据集(来自 R 语言的 datasets 包)给出了从 1962 年第二季度到 1971 年第四季度共 39 条季度收入及其影响因素的数据，y 代表季度收入，lag.quartely.revenue 代表滞后一期的季度收入，price index 为物价指数，income level 为收入水平，market potential 为市场潜力。

(1) 建立一个多元回归模型，用 price index、income level 和 market potential 三个变量解释 y，所有解释变量各自都是统计显著的吗？

(2) 估计当 price index=4.27789，income level=6.20030，market potential=13.1664 时 y 的值。

(3) 预测当 price index=5，income level=7，market potential=14 时 y 的值。

3.3 InsectSprays 数据集(来自 R 语言的 datasets 包)给出了不同种类杀虫剂杀虫效果的数据。count 为杀灭害虫的数量，spray 为不同种类的杀虫剂。问：不同杀虫剂的杀虫效果是否不同？建立一个回归模型进行分析。

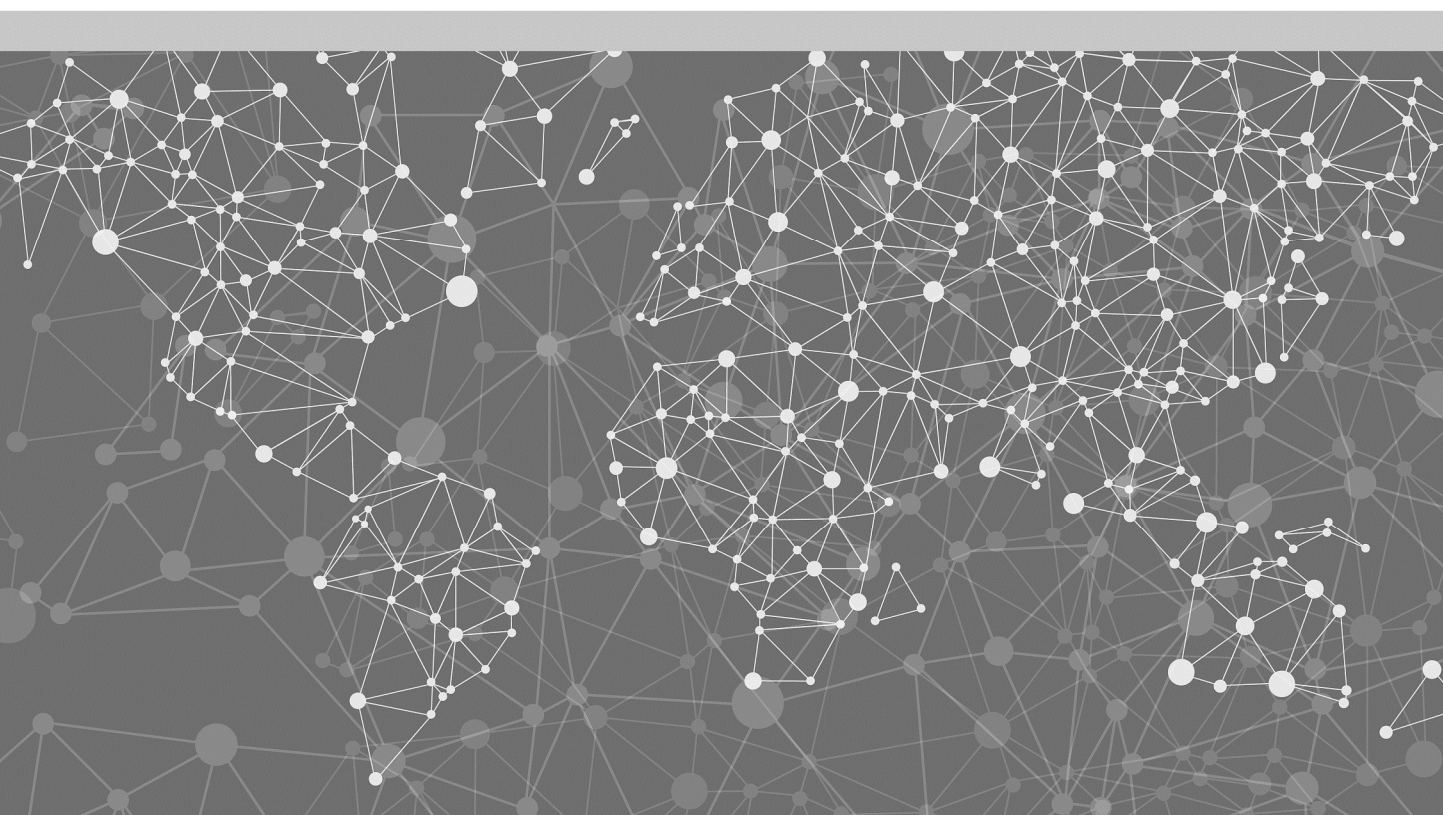
3.4 Loblolly 数据集(来自 R 语言的 datasets 包)给出了松树生长的数据。height 为松树的高度；age 为松树的树龄；seed 为松树的不同树种，它决定了松树能生长的最高高度。

(1) age 是什么类型的变量？如何将它按四分位数转化为有序变量？

(2) 构建 height 与 age、seed 的多元回归模型，分析树龄、树种与树高的关系。

第二部分

数据分析高级方法



第 4 章　多元数据的综合分析

只考虑一个变量(定性或定量)或一个因素(定性变量)对一个观测指标(定量变量)影响大小的问题，称为基本统计分析(如第 2、3 章所述)；考虑一个因素或多个因素对两个或两个以上观测指标(定量变量)的影响大小，或者多个观测指标(定量变量)间的相互关系问题，称为多元统计分析(也称多变量统计分析)。在现实生活中，受多个指标(随机变量)共同作用和影响的现象大量存在。有两种方法可同时对多个随机变量的观测数据进行有效的分析和研究。一种做法是，把多个随机变量分开分析，每次处理一个，逐次分析研究，不过，当变量较多时，变量之间不可避免地存在着相关性，如果分开处理，不仅会丢失很多信息，往往也不容易得到好的研究结论。另一种做法是，同时进行分析研究，即用多元统计分析方法来解决，通过对多个随机变量观测数据的分析，来研究变量之间的相互关系并揭示变量的内在规律。所以说，多元统计分析就是研究多个随机变量之间相互依赖关系及其内在统计规律的一门学科。

例 1.2 给出了 2014 年我国各地区对外贸易国际竞争力数据，显然，这些数据构成了一个多元数据集。下面我们继续使用 pandas 的 read_excel(1)函数读取 Excel 数据。

(1)读取无标签数据

In [1]
```
pd.read_excel('PyDm_data.xlsx','MVdata')
```

Out [1]

	地区	生产总值	从业人员	固定资产	利用外资	进出口额	新品出口	市场占有	对外依存
0	北京	162.519	1069.700	55.789	196.906	3894.9	6470.514	2.635	1.548
1	天津	113.073	763.160	70.677	61.947	1033.9	7490.317	1.986	0.591
2	河北	245.158	3962.420	163.893	178.782	536.0	2288.188	1.276	0.141
3	山西	112.376	1738.900	70.731	104.945	147.6	1522.788	0.242	0.085

……

(2)读取有标签数据

如上所示，在该数据框中，地区只是一个标识，并不参与多元数据分析，这类数据通常需要构建一个分析用的数据框，在参数中增加 index_col=0 即可。

In [2]
```
MVdata=pd.read_excel('PyDm_data.xlsx','MVdata',index_col=0);round(MVdata,3);MVdata
```

Out [2]

地区	生产总值	从业人员	固定资产	利用外资	进出口额	新品出口	市场占有	对外依存
北京	162.519	1069.700	55.789	196.906	3894.9	6470.514	2.635	1.548
天津	113.073	763.160	70.677	61.947	1033.9	7490.317	1.986	0.591
河北	245.158	3962.420	163.893	178.782	536.0	2288.188	1.276	0.141
山西	112.376	1738.900	70.731	104.945	147.6	1522.788	0.242	0.085

……

4.1 多元线性相关与回归

4.1.1 多元线性相关

有时为了书写或建模方便，需要将变量名改为英文或拼音等，比如，在 MVdata 数据框中，变量名都是中文，建模操作有所不便，我们就重新命名变量名。

In [3]:
```
YXdata=pd.read_excel('PyDm_data.xlsx','MVdata',index_col=0);
YXdata.columns=['Y','X1','X2','X3','X4','X5','X6','X7']
YXdata
```

Out [3]:

地区	Y	X1	X2	X3	X4	X5	X6	X7
北京	162.519	1069.700	55.789	196.906	3894.9	6470.514	2.635	1.548
天津	113.073	763.160	70.677	61.947	1033.9	7490.317	1.986	0.591
河北	245.158	3962.420	163.893	178.782	536.0	2288.188	1.276	0.141
山西	112.376	1738.900	70.731	104.945	147.6	1522.788	0.242	0.085
……								

4.1.1.1 多元相关分析

从数学的角度，要研究变量间的关系，通常需要计算其协方差，对多个变量来说，就是计算变量间的协差阵。由于协方差是有单位的，不容易比较，所以通常将其标准化为相关系数，任意两个变量间的相关系数构成的矩阵为

$$\boldsymbol{R}=\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1p} \\ r_{21} & r_{22} & \cdots & r_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ r_{p1} & r_{p2} & \cdots & r_{pp} \end{bmatrix}=\begin{bmatrix} 1 & r_{12} & \cdots & r_{1p} \\ r_{21} & 1 & \cdots & r_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ r_{p1} & r_{p2} & \cdots & 1 \end{bmatrix}=(r_{ij})_{p\times p}$$

式中，r_{ij} 为任意两个变量间的简单相关系数，即 Pearson 相关系数，其计算公式为

$$r_{ij}=\frac{\sum(x_i-\overline{x}_i)(x_j-\overline{x}_j)}{\sqrt{\sum(x_i-\overline{x}_i)^2\sum(x_j-\overline{x}_j)^2}}$$

Perason 相关系数 r_{ij} 的取值范围为[–1,1]，$-1<r_{ij}<0$ 表示具有负线性相关性，其值越接近–1，负相关性越强；$0<r_{ij}<1$ 表示具有正线性相关性，其值越接近 1，正相关性越强；$r_{ij}=-1$ 表示具有完全负线性相关性；$r_{ij}=1$ 表示具有完全正线性相关性；$r_{ij}=0$ 表示两个变量不具有线性相关性。相关系数是协方差的标准化形式，消除了单位的影响。

这里所说的多元相关分析，不是真正意义上的多个变量的相关，只是两个变量相关分析的多元表示，即对多个变量计算两两之间的线性相关系数。下面是上述宏观经济数据的多元相关系数矩阵。

In [4]:
```
Yxdata[['Y','X1']].corr()  #Yxdata['Y'].corr(Yxdata['X1'])
```

Out [4]			
		Y	X1
	Y	1.0000	0.8155
	X1	0.8155	1.0000

In [5]	YXdata.corr()

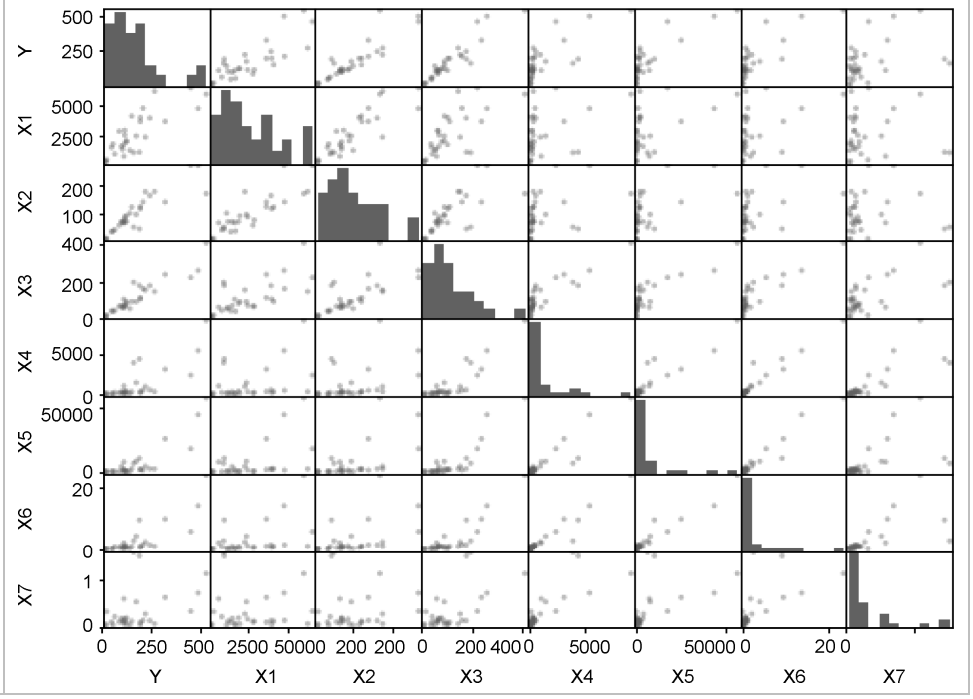
Out [5]								
	Y	X1	X2	X3	X4	X5	X6	X7
Y	1.0000	0.8155	0.8921	0.9287	0.7721	0.8487	0.8116	0.4211
X1	0.8155	1.0000	0.8569	0.6965	0.3865	0.5107	0.4633	0.0107
X2	0.8921	0.8569	1.0000	0.7173	0.4312	0.5802	0.4929	0.0902
X3	0.9287	0.6965	0.7173	1.0000	0.8780	0.8542	0.8692	0.6141
X4	0.7721	0.3865	0.4312	0.8780	1.0000	0.9235	0.9628	0.8097
X5	0.8487	0.5107	0.5802	0.8542	0.9235	1.0000	0.9697	0.5653
X6	0.8116	0.4633	0.4929	0.8692	0.9628	0.9697	1.0000	0.6592
X7	0.4211	0.0107	0.0902	0.6141	0.8097	0.5653	0.6592	1.0000

4.1.1.2 多元数据散点图

下面给出变量两两间的相关系数矩阵散点图。

In [6]	pd.plotting.scatter_matrix(YXdata);

Out [6]



4.1.1.3 相关系数矩阵检验

从前面的相关系数矩阵计算结果可以看出，Y 与各 X 的相关系数都较高，对其所进行的假设检验等同于两两之间的相关系数检验，Python 没有直接产生多个变量两两之间相关系数检验的函数，但可分别进行，如检验 Y 和 X1 之间的线性相关性，可写为 pearsonr(Y,X1)，依此类推，但比较麻烦。可自定义一个函数一次性全部完成检验，下

面调用我们自定义的检验变量间两两相关性的矩阵相关检验函数 mcor_test()，该函数可对相关系数矩阵进行检验。

In [7]
```
dm.mcor_test(YXdata)  #须加载 PyDm_fun 函数包：import PyDm_fun as dm
```

Out [7]

	Y	X1	X2	X3	X4	X5	X6	X7
Y	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0205
X1	0.8155	1.0000	0.0000	0.0000	0.0349	0.0039	0.0099	0.9554
X2	0.8921	0.8569	1.0000	0.0000	0.0173	0.0008	0.0056	0.6353
X3	0.9287	0.6965	0.7173	1.0000	0.0000	0.0000	0.0000	0.0003
X4	0.7721	0.3865	0.4312	0.8780	1.0000	0.0000	0.0000	0.0000
X5	0.8487	0.5107	0.5802	0.8542	0.9235	1.0000	0.0000	0.0011
X6	0.8116	0.4633	0.4929	0.8692	0.9628	0.9697	1.0000	0.0001
X7	0.4211	0.0107	0.0902	0.6141	0.8097	0.5653	0.6592	1.0000

下三角为相关系数，上三角为概率

4.1.2 多元线性回归模型

4.1.2.1 多元线性回归模型形式

在 3.3.2 节中介绍了单变量线性回归分析，它研究的是一个因变量与一个自变量间呈线性趋势的数量关系。在实际中，常会遇到研究一个因变量与多个自变量间数量关系的问题，如在例 1.2 中，考察国内生产总值与其他经济变量间的依存关系，这时需要建立多元回归模型。与一元线性回归(直线回归)类似，一个因变量与多个自变量间的这种线性数量关系可以用多元线性回归方程来表示。

$$y=\beta_0+\beta_1x_1+\beta_2x_2+\cdots+\beta_px_p+\varepsilon$$

式中，β_0 相当于直线回归方程中的常数项，β_i (i=1,2,$\cdots$,p) 称为偏回归系数(partial regression coefficient)，其意义与直线回归方程中的回归系数相似。当其他自变量对因变量的线性影响固定时，β_i 反映第 i 个自变量 x_i 对因变量 y 线性影响程度的大小。这样的回归称为因变量 y 在这组自变量 x 上的回归，习惯称为多元线性回归模型。

(1) 多元线性回归模型的一般形式

随机变量 y 与一般变量 x 的线性回归模型为

$$y=\beta_0+\beta_1x_1+\beta_2x_2+\cdots+\beta_px_p+\varepsilon$$

假设得到 n 组观测数据 $(x_{i1},x_{i2},\cdots,x_{ip},y_i)(i=1,2,\cdots,n)$，将其写成矩阵形式：

$$\boldsymbol{Y}=\boldsymbol{X\beta}+\boldsymbol{\varepsilon}$$

式中，

$$\boldsymbol{Y}=\begin{bmatrix}y_1\\y_2\\\cdots\\y_n\end{bmatrix},\quad \boldsymbol{X}=\begin{bmatrix}1&x_{11}&\cdots&x_{1p}\\1&x_{21}&\cdots&x_{2p}\\\cdots&\cdots&\cdots&\cdots\\1&x_{n1}&\cdots&x_{np}\end{bmatrix},\quad \boldsymbol{\beta}=\begin{bmatrix}\beta_0\\\beta_1\\\cdots\\\beta_p\end{bmatrix},\quad \boldsymbol{\varepsilon}=\begin{bmatrix}\varepsilon_1\\\varepsilon_2\\\cdots\\\varepsilon_n\end{bmatrix}$$

通常称 $\boldsymbol{X}$ 为设计阵，$\boldsymbol{\beta}$ 为回归系数向量。

(2)线性回归模型的基本假设

由于一元线性回归比较简单，其趋势图可用散点图直观显示，所以，我们对其性质和假定并未进行详细探讨。实际上，在建立线性回归模型前，需要对模型做一些假定，经典线性回归模型的基本假设前提如下。

① 一般来说，解释变量是非随机变量。

② 误差等方差及不相关假定(G-M 条件)：

$$\begin{cases} E(\varepsilon_i)=0 & (i=1,2,\cdots,n) \\ \mathrm{Cov}(\varepsilon_i,\varepsilon_j)=\begin{cases}\sigma^2, & i=j \\ 0, & i\neq j\end{cases} & (i,j=1,2,\cdots,n) \end{cases}$$

③ 误差正态分布的假定条件：

$$\varepsilon_i \overset{\text{iid}}{\sim} N(0,\sigma^2) \quad (i=1,2,\cdots,n)$$

④ $n>p$，即要求样本容量个数多于解释变量的个数。

4.1.2.2 多元线性回归参数估计

由多元线性模型的矩阵形式 $\boldsymbol{Y}=\boldsymbol{X\beta}+\boldsymbol{\varepsilon}$ 可知，若模型的参数 $\boldsymbol{\beta}$ 的估计量 $\hat{\boldsymbol{\beta}}$ 已获得，则 $\hat{\boldsymbol{Y}}=\boldsymbol{X}\hat{\boldsymbol{\beta}}$，于是残差 $e_i=y_i-\hat{y}_i$，根据最小二乘原理，所选择的估计方法应使估计值 $\hat{y}_i$ 与观察值 y_i 之间的残差 e_i 在所有样本点上达到最小，即使

$$Q=\sum_{i=1}^{n}(y_i-\hat{y}_i)^2=\boldsymbol{e}'\boldsymbol{e}=(\boldsymbol{Y}-\boldsymbol{X}\hat{\boldsymbol{\beta}})'(\boldsymbol{Y}-\boldsymbol{X}\hat{\boldsymbol{\beta}})$$

达到最小，根据微积分求极值的原理，$\boldsymbol{Q}$ 对 $\hat{\boldsymbol{\beta}}$ 求导且等于 0，可求得使 Q 达到最小的 $\hat{\boldsymbol{\beta}}$，这就是所谓的普通最小二乘(OLS)法。

$$\hat{\boldsymbol{\beta}}=(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}$$

在例 1.2 中，我们发现【生产总值】(Y)与【从业人员】(X1)之间的确存在线性回归关系，为了进一步考察它们和其他变量之间的数量关系，需要建立多元线性回归方程，步骤如下。

(1)建立一个自变量的线性回归模型

In [8]	M1=smf.ols('Y～X1',YXdata).fit(); M1.params
Out [8]	Intercept 14.684200 X1 0.060334

(2)建立两个自变量的线性回归模型

In [9]	M2=smf.ols('Y～X1+X2',YXdata).fit(); M2.params
Out [9]	Intercept −12.451052 X1 0.014231 X2 1.459890

(3)建立三个自变量的线性回归模型

In [10]	M3=smf.ols('Y～X1+X2+X3',YXdata).fit(); M3.params	
Out [10]	Intercept	−23.923166
	X1	0.000715
	X2	0.920087
	X3	0.885195

(4)建立所有自变量的线性回归模型

In [11]	Ms=smf.ols('Y～X1+X2+X3+X4+X5+X6+X7',YXdata).fit(); Ms.params	
Out [11]	Intercept	−7.138064
	X1	0.008497
	X2	0.998300
	X3	0.238449
	X4	0.006917
	X5	0.000906
	X6	4.046515
	X7	10.260713

4.1.2.3 多元线性回归模型检验

由样本资料建立回归方程的目的是对变量间的回归关系进行统计推断，也就是对总体回归方程进行参数估计和假设检验。由样本计算得到的这些偏回归系数是总体偏回归系数的估计值，如果这些总体偏回归系数等于 0，多元回归方程就没有意义，所以，与直线回归一样，在建立方程后有必要对这些偏回归系数进行检验。对多元回归方程进行假设检验也可以用方差分析。

前面对回归模型的系数进行了估计，下面对回归系数进行假设检验。

(1)回归系数的假设检验

多元回归方程有统计学意义，并不说明每个偏回归系数都有意义，所以有必要对每个偏回归系数进行检验。在 $\beta_j=0$ 时，偏回归系数 $\hat{\beta}_j$ (j=1,2,⋯,p)服从正态分布，所以可用 t 统计量对偏回归系数进行检验。

检验假设 $H_{0j}:\beta_j=0$， $H_{1j}:\beta_j\neq 0$。

当 H_{0j} 成立时， $\boldsymbol{\beta}\sim N(\boldsymbol{\beta},\sigma^2(\boldsymbol{X}'\boldsymbol{X})^{-1})$，则构造的 t 统计量为

$$t_j=\frac{\hat{\beta}_j-\beta_j}{s_{\hat{\beta}_j}}\quad (j=1,2,\cdots,p)$$

式中， $s_{\hat{\beta}_j}$ 是第 j 个偏回归系数的标准误。

当原假设 $H_{0j}:\beta_j=0$ 成立时，上面的 t 统计量服从自由度为 $n-p-1$ 的 t 分布。给定显著性水平 α，查出双侧检验的临界值 $t_{1-\alpha/2}$。当 $|t_j|\geqslant t_{1-\alpha/2}$ 时，拒绝原假设 $H_{0j}:\beta_j=0$，认为 β_j 显著不为零，自变量 x_j 对因变量 $\boldsymbol{y}$ 的线性效果显著；当 $|t_j|<t_{1-\alpha/2}$ 时，接受原假设 H_{0j}，认为 β_j 为零，自变量 x_j 对因变量 $\boldsymbol{y}$ 的线性效果不显著。

(2)回归方程的假设检验

方差分析的原假设是 $H_0:\beta_1=\beta_2=\cdots=\beta_p=0$，这就意味着因变量 $\boldsymbol{y}$ 与所有的自变量 x_j 都不存在线性回归关系，多元回归方程没有意义。相应的备择假设 $H_1:\beta_1$，β_2，…，β_p 不全为 0。

由于因变量 $\boldsymbol{y}=\hat{\boldsymbol{y}}+\boldsymbol{e}$，即 $\boldsymbol{y}$ 包含拟合值和误差。因变量 $\boldsymbol{y}$ 的离均差平方和可分解成两部分，即

$$\mathrm{SS}_T=\sum_{i=1}^{n}(y_i-\overline{y})^2=\sum_{i=1}^{n}(\hat{y}_i-\overline{y})^2+\sum_{i=1}^{n}(y_i-\hat{y}_i)^2=\mathrm{SS}_R+\mathrm{SS}_E$$

方差分析的目的是检验回归的变异(方差或均方)是否远大于误差的变异(方差或均方)，如果误差的变异远大于回归的变异，就意味着因变量 $\boldsymbol{y}$ 与自变量 $\boldsymbol{x}$ 不存在依存关系，回归方程没有统计意义。由离均差平方和可计算回归的均方(方差)$\mathrm{MS}_R=\mathrm{SS}_R/p$ 和误差的均方(方差)$\mathrm{MS}_E=\mathrm{SS}_E/(n-p-1)$。

进而计算方差分析的 F 值

$$F=\frac{\mathrm{MS}_R}{\mathrm{MS}_E}\sim F(p,n-p-1)$$

这里 F 服从自由度为 p 和 $n-p-1$ 的 F 分布，这样就可以用 F 统计量来检验回归方程是否有意义了。

如果有确切的回归线，那么误差项和残差就是一致的，故此时可利用残差值来估计 σ 的值。

$$S_e=\sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}=\sqrt{\frac{\mathrm{SS}_E}{n-2}}$$

这里 S_e 是 σ 的无偏估计，称为剩余标准差或剩余标准误(residual standard error)，它反映了因变量 $\boldsymbol{y}$ 在扣除自变量 $\boldsymbol{x}$ 的线性影响后的离散程度。s_e 可以与 $\boldsymbol{y}$ 的标准差 s_y 比较，从而可看出自变量 $\boldsymbol{x}$ 对 $\boldsymbol{y}$ 的线性影响的大小。

一般统计软件在完成多元回归分析的同时都会输出方差分析与 t 检验的结果，其中 t 检验结果给出每个偏回归系数和常数项的值、标准误、t 值与相应的 P 值。

In [12]	M1.summary()			
Out [12]	OLS Regression Results			
	==========			
	Dep. Variable:	Y	R-squared:	0.665
	Model:	OLS	Adj. R-squared:	0.653
	Method:	Least Squares	F-statistic:	55.61
	Date:	Wed, 21 Mar 2018	Prob (F-statistic):	4.02e−08
	Time:	14:20:37	Log-Likelihood:	−171.88
	No. Observations:	30	AIC:	347.8
	Df Residuals:	28	BIC:	350.6

	Df Model: 1 Covariance Type: nonrobust ========== coef std err t P>|t| [0.025 0.975] ---------- Intercept 14.6842 25.540 0.575 0.570 −37.632 67.000 X1 0.0603 0.008 7.457 0.000 0.044 0.077 ========== Omnibus: 2.833 Durbin-Watson: 1.612 Prob（Omnibus）: 0.243 Jarque-Bera（JB）: 2.029 Skew: 0.637 Prob（JB）: 0.363 Kurtosis: 3.031 Cond. No. 5.73e+03 ==========

由假设检验结果可见，模型的 $P = 4.02e{-}08 << 0.05$，认为回归模型有意义。由 t 检验结果可见，偏回归系数的 P 值小于 0.05，可认为【从业人员】(X1)对【生产总值】有显著影响。

In [13]	M2.summary()
Out [13]	OLS Regression Results ========== Dep. Variable: Y R-squared: 0.806 Model: OLS Adj. R-squared: 0.791 Method: Least Squares F-statistic: 55.97 Date: Wed, 21 Mar 2018 Prob（F-statistic）: 2.48e−10 Time: 14:21:39 Log-Likelihood: −163.72 No. Observations: 30 AIC: 333.4 Df Residuals: 27 BIC: 337.6 Df Model: 2 Covariance Type: nonrobust ========== coef std err t P>|t| [0.025 0.975] ---------- Intercept −12.4511 20.742 −0.600 0.553 −55.010 30.108 X1 0.0142 0.012 1.169 0.253 −0.011 0.039 X2 1.4599 0.330 4.419 0.000 0.782 2.138 ========== Omnibus: 25.116 Durbin-Watson: 1.910 Prob（Omnibus）: 0.000 Jarque-Bera（JB）: 43.075 Skew: 1.941 Prob（JB）: 4.43e−10 Kurtosis: 7.403 Cond. No. 6.00e+03 ==========

由假设检验结果可见，模型的 P=2.48e−10<0.001，认为该回归模型也有意义。由 t 检验结果可见，【从业人员】(X1)的偏回归系数的 P 值大于 0.05，而【固定资产】(X2)

的 P 值小于 0.05，可认为【从业人员】对【生产总值】的影响不大，而【固定资产】对【生产总值】有较大影响。

In [14]
```
M3.summary()
```

Out [14]
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      Y   R-squared:                       0.968
Model:                            OLS   Adj. R-squared:                  0.964
Method:                 Least Squares   F-statistic:                     259.2
Date:                Wed, 21 Mar 2018   Prob (F-statistic):           1.76e-19
Time:                        14:22:18   Log-Likelihood:                -136.83
No. Observations:                  30   AIC:                             281.7
Df Residuals:                      26   BIC:                             287.3
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    -23.9232      8.684     -2.755      0.011     -41.772      -6.074
X1             0.0007      0.005      0.138      0.892      -0.010       0.011
X2             0.9201      0.145      6.333      0.000       0.621       1.219
X3             0.8852      0.078     11.408      0.000       0.726       1.045
==============================================================================
Omnibus:                        3.524   Durbin-Watson:                   2.149
Prob(Omnibus):                  0.172   Jarque-Bera (JB):                2.910
Skew:                          -0.758   Prob(JB):                        0.233
Kurtosis:                       2.829   Cond. No.                     6.05e+03
==============================================================================
Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.05e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

由假设检验结果可见，模型的 P=1.76e–19<0.001，认为该回归模型也有意义。由 t 检验结果可见，【从业人员】(X1)的偏回归系数的 P 值大于 0.05，而【固定资产】(X2)和【利用外资】(X3)的 P 值小于 0.05，可认为【固定资产】和【利用外资】对【生产总值】有较大影响，而【从业人员】对【生产总值】的影响不大。

In [15]
```
Ms.summary()
```

Out [15]
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      Y   R-squared:                       0.991
Model:                            OLS   Adj. R-squared:                  0.988
Method:                 Least Squares   F-statistic:                     329.6
```

Date:	Wed, 21 Mar 2018	Prob（F-statistic）:	9.08e–21	
Time:	14:22:38	Log-Likelihood:	–118.36	
No. Observations:	30	AIC:	252.7	
Df Residuals:	22	BIC:	263.9	
Df Model:	7			
Covariance Type:	nonrobust			

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	–7.1381	7.243	–0.985	0.335	–22.159	7.883
X1	0.0085	0.004	2.277	0.033	0.001	0.016
X2	0.9983	0.107	9.354	0.000	0.777	1.220
X3	0.2384	0.127	1.879	0.074	–0.025	0.502
X4	0.0069	0.012	0.566	0.577	–0.018	0.032
X5	0.0009	0.001	0.748	0.462	–0.002	0.003
X6	4.0465	3.436	1.178	0.251	–3.079	11.172
X7	10.2607	23.487	0.437	0.666	–38.449	58.970

Omnibus:	1.261	Durbin-Watson:	2.370
Prob（Omnibus）:	0.532	Jarque-Bera（JB）:	0.425
Skew:	–0.231	Prob（JB）:	0.809
Kurtosis:	3.355	Cond. No.	1.34e+05

Warnings:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The condition number is large, 1.34e+05. This might indicate that there are strong multicollinearity or other numerical problems.

由假设检验结果可见，模型的 p=9.08e–21<0.001，认为该回归模型也有意义。由 t 检验结果可见，【从业人员】(X1)和【固定资产】(X2)的偏回归系数的 P 值小于 0.05，其他变量的 P 值大于 0.05，可认为【从业人员】和【固定资产】对【生产总值】的影响较大，其他变量影响较小。

从前面的分析也可以看到，模型的建立是一个复杂的过程，需要研究者不断探索，以获得较为有用的模型。

4.1.2.4 多元线性回归模型评判

在建立回归模型时，要求误差服从独立同正态分布，即

$$\varepsilon_i \overset{\text{iid}}{\sim} N(0,\sigma^2) \quad (i=1,2,\cdots,n)$$

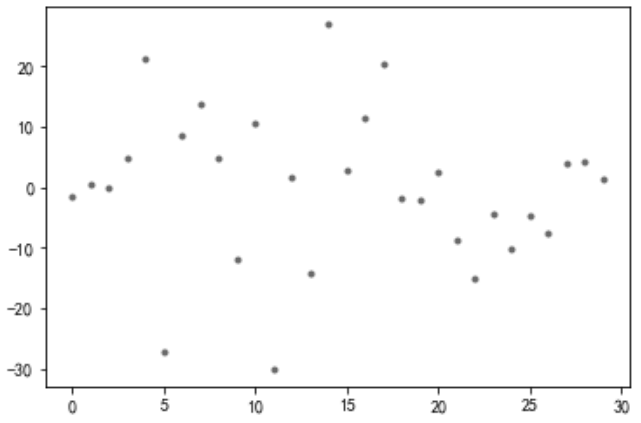
(1)误差的相关性验证

对于多元回归模型，如果随机误差项的各期值之间存在相关关系，即

$$\mathrm{Cov}(u_t,u_s)=E(u_tu_s)\neq 0 \quad (t\neq s;\ \ t,s=1,2,\cdots,k)$$

则称随机误差项之间存在自相关性(autocorrelation)。

对其验证的最直观方法就是看残差的分布图，如下所示。

In [16]	et=Ms.resid　　#模型 Ms 的残差 et plt.plot(et,'.');　　#残差图
Out [16]	

从残差的序列图可以看出，该模型的误差有可能不存在相关性。

(2)误差自相关性检验

若设误差的自相关系数为 ρ，则其样本估计公式为

$$\hat{\rho}=\frac{\sum_{t=2}^{n}(e_t-\overline{e}_t)(e_{t-1}-\overline{e}_{t-1})}{\sqrt{\sum_{t=1}^{n}(e_t-\overline{e}_t)^2\sum_{t=2}^{n}(e_{t-1}-\overline{e}_{t-1})^2}}=\frac{\sum_{t=2}^{n}e_te_{t-1}}{\sqrt{\sum_{t=1}^{n}e_t^{\,2}\sum_{t=2}^{n}e_{t-1}^2}}$$

式中，e_t 表示残差序列，e_{t-1} 表示残差的滞后一阶序列。

In [17]	ro=et.corr(et.shift(1));ro　　#et.shift(1) =et−1
Out [17]	−0.18553688673392713

残差相关系数很小，说明误差的自相关性不大。

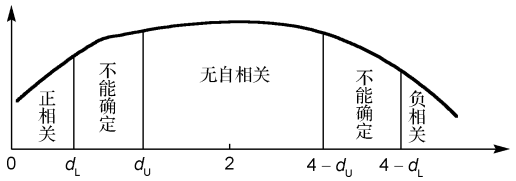
检验模型残差是否存在一阶自相关性最常用的方法是 Durbin 检验，Durbin 和 Watson 于 1951 年提出一种检测序列自相关的方法，即 D-W 检验。

Durbin 和 Watson 针对原假设 H_0: $\rho=0$，即不存在一阶自回归，构造如下统计量：

$$\mathrm{DW}=\frac{\sum_{t=2}^{n}(e_t-e_{t-1})^2}{\sum_{t=1}^{n}e_t^2}$$

该统计量的分布与出现在给定样本中的 X 值有复杂的关系，因此其精确的分布很难得到，但是，Durbin 和 Watson 成功导出了临界值的下限 d_{L} 和上限 d_{U}，且这些上、下限

只与样本的容量 n 和解释变量的个数 k 有关，而与解释变量 X 的取值无关，下图是 DW 相关性临界值的示意图。



显然， $DW \approx 2(1-\hat{\rho})$，由此也可得到 $\hat{\rho} \approx 1-DW/2$。

如果存在完全一阶正相关，即 ρ=1，则 DW≈ 0；

如果存在完全一阶负相关，即 ρ= –1，则 DW≈ 4；

如果存在完全不相关，即 ρ= 0，则 DW≈2。

对被估计模型的残差进行 Durbin-Watson 检验，结果见 4.1.2.3 节的 Ms 模型的检验表。

In [18]	Ms.summary2().tables[2][3][0]　　#DW 值 DW=2*(1-ro);DW
Out [18]	2.3710737734678542

DW 接近 2，说明模型 Ms 的残差不存在一阶自相关。

也可用以下函数直接计算 DW 值：

In [19]	sm.stats.durbin_watson(Ms.resid)
Out [19]	2.3710737734678542

(3) 误差的正态性检验

Jarque-Bera 检验：检验序列是否符合正态分布的一种正态性检验方法。当序列服从正态分布时，JB 统计量：

$$JB = n\left(\frac{skew^2}{6} + \frac{kurt^2}{24}\right)$$

渐进服从 $\chi^2(2)$ 分布。式中，n 为样本规模，skew、kurt 分别为随机变量的偏度和峰度。结果见 4.1.2.3 节的 Ms 模型的检验表。

In [20]	Ms.summary2().tables[2][3][1]　　#JB 值
Out [20]	'0.425'
In [21]	Ms.summary2().tables[2][3][2]　　#JB 概率
Out [21]	'0.809'

从中可知，该模型的残差基本服从正态分布 (p=0.809>0.05)。

(4) 模型的决定系数

在实际分析中，一个变量的变化往往受多种变量的综合影响，这就需要采用决定系数来判断模型的好坏程度。

决定系数实际就是回归离差平方和与总离差平方和的比值，反映了回归贡献的百分

比值，所以常把 R^2 称为模型的决定系数。

$$R^2=\frac{\mathrm{SS}_R}{\mathrm{SS}_T}$$

R^2 在模型评价、变量选择、衡量曲线回归方程拟合的好坏程度时常用。

In [22]	R2=Ms.summary2().tables[0][1][6];R2　　#模型的决定系数 R2
Out [22]	'0.991'

在应用过程中发现，如果在模型中增加一个解释变量，R^2 往往增大。这就给人一种错觉：要使模型拟合得好，只要增加解释变量即可。而现实情况往往是，由增加解释变量个数引起的 R^2 的增大与拟合好坏无关，R^2 需要调整。于是就有了调整的可决系数 $\mathrm{adj}R^2$（adjusted coefficient of determination）。在样本容量一定的情况下，增加解释变量必定使得自由度减少，所以调整的思路是：将残差平方和与总离差平方和分别除以各自的自由度，以剔除变量个数对模型的影响:

$$\mathrm{adj}R^2=1-\frac{\mathrm{SS}_R/(n-p-1)}{\mathrm{SS}_T/(n-1)}$$

In [23]	Ms.summary2().tables[0][3][0]　　#adj.R2
Out [23]	'0.988'

(5) 多元复相关系数

复相关系数用来判断因变量和多个自变量之间线性拟合的程度。

设因变量为 y，自变量为 $x_1,x_2,\cdots,x_p$，对 y 与 $x_1,x_2,\cdots,x_p$ 的多元相关就是对 y 与其拟合值 $\hat{y}$ 的相关，记 $R=r_{y\cdot x_1x_2\ldots x_p}$ 为 y 与 $x_1,x_2,\cdots,x_p$ 的复相关系数，计算公式为

$$R=r_{y\cdot x_1x_2\cdots x_p}=r_{y\cdot\hat{y}}=\frac{\mathrm{Cov}(y,\hat{y})}{\hat{\sigma}_y\hat{\sigma}_{\hat{y}}}=\frac{\sigma_{\hat{y}}}{\hat{\sigma}_y}=\sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i-\bar{y})^2}{\sum_{i=1}^{n}(y_i-\bar{y})^2}}=\sqrt{\frac{\mathrm{SS}_R}{\mathrm{SS}_T}}$$

复相关系数反映了一个变量与另一组变量关系密切的程度。复相关系数的假设检验等价于多元回归的方差分析结果，所以不必再进行假设检验。

In [24]	from math import sqrt R=sqrt(float(R2));R
Out [24]	0.9954898291795853

模型总结：线性回归模型的结果表明，国内生产总值与其他经济指标之间存在显著的相关关系，其中，可决系数为 0.991，F 统计量和 Omnibus 统计量的 P 值都接近于 0，表明模型拟合效果较好。Durbin-Watson 检验的值为 2.371，表明残差数据不存在序列相关性。Jarque-Bera 的 P 值接近于 0，表明误差数据服从正态分布。有时为了方便，也可用表格的形式对回归模型进行比较。

In [25]	from statsmodels.iolib.summary2 import summary_col summary_col([M1,M2,M3,Ms]) #模型结果比较

Out [25]

```
==============================================
                Y I       Y II      Y III     Y IIII
----------------------------------------------
Intercept     14.6842   -12.4511   -23.9232   -7.1381
              (25.5399) (20.7420)  (8.6835)  (7.2431)
X1            0.0603    0.0142     0.0007     0.0085
              (0.0081)  (0.0122)   (0.0052)   (0.0037)
X2                      1.4599     0.9201     0.9983
                        (0.3304)   (0.1453)   (0.1067)
X3                                 0.8852     0.2384
                                   (0.0776)   (0.1269)
X4                                            0.0069
                                              (0.0122)
X5                                            0.0009
                                              (0.0012)
X6                                            4.0465
                                              (3.4358)
X7                                            10.2607
                                              (23.4872)
==============================================
Standard errors in parentheses.
```

4.2 综合评价方法

4.2.1 综合评价指标体系

4.2.1.1 评价指标体系的构建

在现实生活中，对一些事物的分析和评价常常涉及多个因素或多个指标，评价是在多个因素相互作用下的一种综合判断。比如，要判断哪个企业的绩效好，就得从若干企业的财务管理、销售管理、生产管理、人力资源管理、研究与开发能力等多方面进行综合比较；要了解全国各地区的知识产权发展情况，就得从全国各地区的专利发展情况、商标发展情况、版权发展情况，以及其他方面发展情况等多个方面进行综合比较；等等。因此，可以这样说，几乎所有的综合性活动都可以进行综合评价，而且不能只考虑被评价对象的某一个方面，而必须全面地从整体的角度对被评价对象进行评价。

多指标综合评价方法具有以下特点：包含若干指标，分别说明被评价对象的不同方面；评价方法最终要对被评价对象做出一个整体性的评判，用一个总指标来说明被评价对象的一般水平。

在多指标综合评价中，评价指标体系的构建是最重要的问题，是综合评价能准确反映全面情况的前提。如果评价指标选择不当，再好的综合评价方法也会出现差错，甚至

完全失败。构建综合评价指标体系应遵循以下几项原则：

① 系统全面性原则。例如，在经济社会发展水平的评价中，综合评价指标体系必须能够较全面地反映经济社会发展的综合水平，指标体系应包括经济水平、科技进步、社会发展和生态环境等各个主要方面的内容。除了设置上述指标外，还应考虑设置与之关系密切的经济结构、人口素质、居民物质生活水平和自然资源等指标。

② 稳定可比性原则。综合评价指标体系中选用的指标既要有稳定的数据来源，又要适应我国实际状况，指标的统计口径(包括指标的时间长度、计量单位、内容含义)必须一致可比，才能保证评估结果的真实、客观和合理。

③ 简明科学性原则。在系统全面性的基础上，尽量选择具有代表性的综合指标，要避免选择含义相近的指标。指标体系中指标的多少须适宜，指标体系的设置应具有一定的科学性，既简明又科学。

④ 灵活可操作性原则。综合评价指标体系在实际应用中应具有一定的灵活性，以方便全国各地区不同发展水平、不同层次评价对象的操作使用。各个指标的数据来源渠道要畅通，具有较强的操作性。

例如，我国各地区对外贸易国际竞争力情况的指标体系如表 4-1 所列，数据参见例 1.2，其中变异系数法求权重见下一节。

表 4-1　我国各地区对外贸易国际竞争力情况的指标体系

	指　　标	权重(等权)	权重(变异系数法)
区域对外贸易国际竞争力指标体系	生产总值	1/8	0.078960
	从业人员	1/8	0.070345
	固定资产	1/8	0.067101
	利用外资	1/8	0.079711
	进出口额	1/8	0.177985
	新品出口	1/8	0.205693
	市场占有	1/8	0.191693
	对外依存	1/8	0.128513

4.2.1.2　评价指标的基本分析

续例 1.2，我国各地区对外贸易国际竞争力的单变量分析。

下面对我国各地区对外贸易国际竞争力数据进行单变量统计分析，首先对各地区国内生产总值进行排名，由于这时是单指标，故可直接对其数据进行排序。

In　[26]	GDP=pd.DataFrame(MVdata['生产总值']);GDP GDP['排序']=(-GDP).rank(); GDP　　#GDP['排序']=GDP.rank(ascending=False)
Out [26]	生产总值　排序 地区 北京　162.5193　13.0 天津　113.0728　20.0

	河北	245.1576	6.0
	山西	112.3755	21.0
	……		
	甘肃	50.2037	27.0
	青海	16.7044	30.0
	宁夏	21.0221	29.0
	新疆	66.1005	25.0

这里参数 ascending=True 表示从大到小排序(编秩)，也可以用“–”来表示。

Python 可直接对数据框中各变量一次排序，下面对每个变量进行综合排名。

In [27]	(-MVdata).rank()　#MVdata.rank(ascending=False)							
Out [27]	生产总值	从业人员	固定资产	利用外资	进出口额	新品出口	市场占有	对外依存
	地区							
	北京	13.0	25.0	23.0	5.0	4.0	8.0	7.0	1.0
	天津	20.0	27.0	21.0	22.0	8.0	7.0	9.0	6.0
	河北	6.0	8.0	6.0	7.0	10.0	12.0	11.0	16.0
	山西	21.0	19.0	20.0	13.0	23.0	16.0	23.0	23.0
	……								
	甘肃	27.0	21.0	27.0	26.0	27.0	25.0	28.0	21.0
	青海	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
	宁夏	29.0	29.0	29.0	29.0	29.0	27.0	29.0	26.0
	新疆	25.0	26.0	25.0	25.0	19.0	29.0	18.0	11.0

该方法不适于对多变量数据进行综合排序，因为数据之间单位和量纲有可能不同，无法直接相加，故而也就无法进行综合评价。

4.2.2　综合评价分析方法

4.2.2.1　指标的无量纲化

虽然 MVdata 的所有变量都是计量数据，但显然这些变量的单位和量纲还是不同的，通常需要将它们进行无量纲化转换。观测指标的无量纲化指通过某种变换方式消除各个观测指标的计量单位，使其统一、可比的变换过程。常用的无量纲化处理方法主要有以下几种。

(1)标准化变换方法

$$z_{ij}=\frac{x_{ij}-\bar{x}_j}{s_j}\quad (i=1,2,\cdots,n;\ j=1,2,\cdots,p)$$

式中，x_{ij} 是观测值，$\bar{x}_j$ 是均值，s_j 是标准差。经过标准化变换后的指标 z_{ij}，其全部 n 个个体的均值为 0，方差为 1。由于标准差的计量单位与观测值变量本身的计量单位相同，所以变换后的指标不再具有计量单位。

(2)规格化变换方法

$$z_{ij}=\frac{x_{ij}-x_{j\min}}{x_{j\max}-x_{j\min}}\quad (i=1,2,\cdots,n;\ j=1,2,\cdots,p)$$

式中，x_{ij} 是观测值，$x_{j\min}$ 是第 j 个指标的最小观测值，$x_{j\max}$ 是第 j 个指标的最大观测值。经过规格化变换，消除了观测值的计量单位，变换后的指标 z_{ij} 值都在 0～1 之间。

在实际变换中，人们习惯于按百分制对所评价总体中的各个观察单位进行变换，常将上述变换公式乘以 100。此外，有时为使综合评价指标不出现 0 和负值，常在变换公式后加一个常数项，其改进的无量纲方法如下：

$$z_{ij}=\frac{x_{ij}-x_{j\min}}{x_{j\max}-x_{j\min}}\cdot b+a$$

通过这种变换，可使数据限定在[a,b]之间变化，使得数值可比，常取 a=40，b=60。

续例 1.2，我国各地区对外贸易国际竞争力数据的无量纲化。

① 标准化变换。

下面应用 Python 强大的 apply 函数对每列数据进行标准化变换。

In [28]:
```
def bz(x): return (x-x.mean())/x.std()
BZ=MVdata.apply(bz,0);BZ
```

Out [28]:

地区	生产总值	从业人员	固定资产	利用外资	进出口额	新品出口	市场占有	对外依存
北京	−0.085	−0.884	−0.705	0.931	1.300	−0.020	−0.036	3.031
天津	−0.463	−1.057	−0.477	−0.608	−0.087	0.057	−0.162	0.653
河北	0.547	0.751	0.953	0.724	−0.329	−0.336	−0.299	−0.464
山西	−0.468	−0.506	−0.476	−0.117	−0.517	−0.394	−0.499	−0.604
内蒙古	−0.229	−0.783	0.029	−0.693	−0.531	−0.483	−0.506	−0.682
辽宁	0.372	−0.152	1.158	0.457	−0.123	−0.196	−0.105	−0.122
……								

② 规范化变量，计算各个指标的单向评价分数，取 $a=0$，$b=1$。

这种无量纲方法的好处是，它不仅在纵向上消除了不同指标的不同数量级的影响，在横向上还能使得各地区的得分处于 0～1 之间，易于比较，计算结果如下。

In [29]:
```
def gf(x): return (x-x.min())/(x.max()-x.min())
GF=MVdata.apply(gf,0);GF
```

Out [29]:

地区	生产总值	从业人员	固定资产	利用外资	进出口额	新品出口	市场占有	对外依存
北京	0.283	0.123	0.164	0.466	0.426	0.114	0.110	1.000
天津	0.187	0.074	0.222	0.129	0.112	0.132	0.083	0.367
河北	0.443	0.591	0.591	0.421	0.058	0.040	0.053	0.070
山西	0.186	0.231	0.223	0.236	0.015	0.027	0.009	0.033
内蒙古	0.246	0.152	0.353	0.110	0.012	0.006	0.008	0.012
辽宁	0.399	0.333	0.644	0.362	0.104	0.073	0.095	0.161
……								

把数据无量纲化之后，在纵向上数据对比清晰，便于理解分析。

评价指标的合成方法指无量纲化变换后的各个指标按照某种方法进行综合，得出一个可用于评价比较的综合指标。综合评价方法较多，如综合评分法、综合指数法、秩和比法、层次分析法等几种具有代表性的评价方法，这里只介绍一些常用的简单方法。

4.2.2.2 简单平均评价法

简单平均评价法的计算方法是把各指标的得分直接相加，得到一个总分，然后除以指标个数，最后根据这个平均得分的高低来判定评价对象的优劣。这种方法的好处是，对各指标赋予同样的权重来同等看待，省去了确定指标权重的复杂步骤，是最简单的综合评分法。

$$S_i = \sum_{j=1}^{m} w_j z_{ij} = \sum_{j=1}^{m} \frac{1}{m} z_{ij} = \frac{1}{m} \sum_{j=1}^{m} z_{ij}$$

式中，S_i 是评价总体中第 i 个观察单位的综合评价值，m 是指标个数。

(1) 标准化法

In [30]	#建立得分与排名数据框
	SR=pd.DataFrame();SR
	SR['BZscore']=BZ.mean(axis=1);
	SR['BZrank']=SR.BZscore.rank(ascending=False); SR

Out [30]	BZscore BZrank 地区 北京 0.441438 6.0 天津 –0.268043 15.0 河北 0.193261 8.0 山西 –0.447742 21.0 内蒙古 –0.484715 23.0 辽宁 0.160941 9.0 .....

(2) 规范化法

In [31]	SR['GFscore']=GF.mean(1); SR['GFrank']=SR.GFscore.rank(ascending=False); SR
Out [31]	BZscore BZrank GFscore GFrank 地区 北京 0.441438 6.0 0.335638 6.0 天津 –0.268043 15.0 0.163141 15.0 河北 0.193261 8.0 0.283302 8.0 山西 –0.447742 21.0 0.119917 22.0 内蒙古 –0.484715 23.0 0.112328 23.0 辽宁 0.160941 9.0 0.271239 10.0 .....

位列前 5 位的地区分别为广东、江苏、山东、浙江和上海，其中广东分值最高。位列后 3 位的地区分别为海南、宁夏和青海，详见后面的表 4-2。

4.2.2.3 加权综合评分法

简单算术平均法将不同评价指标的重要性同等看待，但现实中综合评价指标体系各指标的重要性是不同的，故应赋予不同分量的权重，才能准确地反映综合指标的合成值。

采用综合评分法进行计算时，对不同指标给出合适的权重是一个关键的问题，选择不同的权重，很可能会出现不同的评价结果。前面是按照平均法计算的综合得分，从排名中可以清楚地看出每个地区经过平均法计算后的排名，选用其他方法可能会得到不同的综合得分和排名。

(1)评价指标的权重

评价指标的权重指在评价指标体系中每个指标的重要程度占该指标群的比重。在多指标综合评价中，各指标在指标群中的重要性不同，因此，不能等量齐观，必须客观地确定各指标的权重。权重值的确定准确与否直接影响综合评价的结果，因而，科学地确定指标权重在多指标综合评价中具有举足轻重的作用。目前国内外关于多指标综合评价的方法很多，根据权重确定方法的不同，这些方法可以大致分为主观赋权法和客观赋权法两类。德尔菲法是一种主观赋权法，层次分析法是一种半主观、半客观的赋权法，变异系数法和熵值法是两种客观赋权法，给出的指标权重值比德尔菲法和层次分析法有较高的可信度，但对数据要求较高，如正态数据等。主成分法和因子分析法也是一种客观赋权法，但通常会损失一些信息。

① 德尔菲(Delphi)法确定权重。

20 世纪 40 年代，美国兰德公司以德尔菲集会形式，向一组专家征询意见，将专家们对过去历史资料的解释和对未来的分析判断汇总整理，经过多次反馈，尽可能取得统一意见。因此，德尔菲法也称为专家评估法。

在综合评价指标的权重确定中，为了提高权重的准确性，往往需要聘请评价对象所属领域内专家对各个评价指标的重要程度进行评定，给出权重。一般程序是，先由各个专家单独对各个评价指标的重要程度进行评定，然后由综合评价人员对各个专家的评定结果进行综合，计算出平均数，再反馈给各位专家，如此反复进行几次，使各位专家的意见趋于一致，就可以确定出各评价指标的权重。

该方法需要多个专家打分，实际操作比较困难，成本也较高。

② 层次分析法确定权重。

层次分析法计算过程的核心问题是权重的构造。自 1982 年层次分析法引入我国以来，人们不仅将之应用于各种决策分析，也用于综合评价权重的构造。其思路如下：建立评价对象的综合评价指标体系，通过指标之间的两两比较确定各自的相对重要程度，然后通过特征值法、最小二乘法等的客观运算来确定各评价指标权重，其中特征值法是层次分析法中提出最早、使用最广泛的权重构造方法，具体方法参考《多元统计分析及 R 语言建模》(第四版)。

该方法在指标较少时基本适用，但当指标较多时，要给出一个合理的判断矩阵不太容易，所以很难得出一个合理的权重。

③ 变异系数法确定权重。

变异系数又称“标准差率”，是衡量资料中各观测值变异程度的一种统计量。当进行两个或多个资料变异程度的比较时，如果度量单位与平均数相同，可以直接利用标准差来比较；如果单位或均数不同，比较其变异程度就不能采用标准差，而要采用标准差与均数的比值(相对值)来比较。

变异系数法确定权重，直接利用各项指标所包含的信息，通过计算得到指标的权重，是一种客观赋权的方法。此方法的基本做法是，在评价指标体系中，指标取值差异越大的指标，也就是越难以实现的指标，这样的指标更能反映被评价单位的差距。例如，在评价各个国家的经济发展状况时，选择人均国民生产总值(人均 GDP)作为评价的标准指标之一，是因为人均 GDP 不仅能反映各个国家的经济发展水平，还能反映一个国家的现代化程度。

标准差 s 与均数 $\bar{x}$ 的比值称为变异系数，记为 CV。变异系数可以消除单位或平均数不同对两个或多个资料变异程度比较结果的影响，显然这个方法只对计量数据有效，对计数数据通常用层次分析法确定权重。

$$\mathrm{CV}=\frac{s}{\bar{x}}$$

In [32]	CV=MVdata.std()/MVdata.mean();CV	#变异系数
Out [32]	生产总值 0.753930 从业人员 0.671672 固定资产 0.640697 利用外资 0.761106 进出口额 1.699455 新品出口 1.964018 市场占有 1.830338 对外依存 1.227084	
In [33]	W=CV/sum(CV);W	#权重
Out [33]	生产总值 0.078960 从业人员 0.070345 固定资产 0.067101 利用外资 0.079711 进出口额 0.177985 新品出口 0.205693 市场占有 0.191693 对外依存 0.128513	

④ 主成分法确定权重，具体见下一节。

(2)加权评分法

用各指标的得分乘以权重求得各指标对各方案的加权得分，每个方案各指标加权得分之和除以权重所得到的商就是总的加权评分，得分最高的方案就是最佳方案。加权评分法的计算公式是

$$S_i = \sum_{j=1}^{m} w_j z_{ij}$$

式中，z_{ij} 是无量纲化数据，w_j 是第 j 个指标的权重，S_i 是评价总体中第 i 个观察单位的综合评价值，m 是指标个数。

下面按变异系数法确定的权重来计算标准化加权得分。

In [34]
```
SR['CVscore']=np.dot(BZ,W)
SR['CVrank']=SR.CVscore.rank(ascending=False); SR
```

Out [34]

地区	BZscore	BZrank	GFscore	GFrank	CVscore	CVrank
北京	0.441438	6.0	0.335638	6.0	0.567790	6.0
天津	–0.268043	15.0	0.163141	15.0	–0.142378	12.0
河北	0.193261	8.0	0.283302	8.0	–0.027057	10.0
山西	–0.447742	21.0	0.119917	22.0	–0.460289	21.0
内蒙古	–0.484715	23.0	0.112328	23.0	–0.504886	23.0
辽宁	0.160941	9.0	0.271239	10.0	0.034652	8.0
.....						

从简单评分法和加权分析法的结果可以看出，两种计算结果还是有一些差别的，因为综合评分法用的是等权，而加权分析法给出了不同指标的权重，但总的趋势应该差不多。表 4-2 所列是三种标准化的综合得分结果比较。

表 4-2　三种综合评价方法结果比较

地区	标准化得分	标准化排名	规范化得分	规范化排名	变异系数得分	变异系数排名
北京	0.441438	6.0	0.335638	6.0	0.567790	6.0
天津	–0.268043	15.0	0.163141	15.0	–0.142378	12.0
河北	0.193261	8.0	0.283302	8.0	–0.027057	10.0
山西	–0.447742	21.0	0.119917	22.0	–0.460289	21.0
内蒙古	–0.484715	23.0	0.112328	23.0	–0.504886	23.0
辽宁	0.160941	9.0	0.271239	10.0	0.034652	8.0
吉林	–0.528942	24.0	0.100920	24.0	–0.508882	24.0
黑龙江	–0.391729	18.0	0.135642	18.0	–0.401298	18.0
上海	0.639790	5.0	0.379174	5.0	0.891055	5.0
江苏	1.970365	2.0	0.707645	2.0	2.052247	2.0
浙江	1.015813	4.0	0.473956	4.0	1.083671	3.0
安徽	–0.109904	13.0	0.211449	13.0	–0.235364	14.0
福建	0.070623	11.0	0.248853	11.0	0.121499	7.0
江西	–0.342605	16.0	0.149691	16.0	–0.368720	16.0
山东	1.253726	3.0	0.547733	3.0	0.965704	4.0
河南	0.312073	7.0	0.320781	7.0	0.006350	9.0
湖北	–0.078386	12.0	0.217328	12.0	–0.219535	13.0
湖南	–0.109916	14.0	0.210795	14.0	–0.260236	15.0
广东	2.832263	1.0	0.905282	1.0	3.175428	1.0
广西	–0.377179	17.0	0.142459	17.0	–0.415602	19.0

续表

地区	标准化得分	标准化排名	规范化得分	规范化排名	变异系数得分	变异系数排名
海南	–0.789672	28.0	0.035354	28.0	–0.647725	28.0
重庆	–0.432676	20.0	0.124261	20.0	–0.394303	17.0
四川	0.141403	10.0	0.273013	9.0	–0.071767	11.0
贵州	–0.673789	26.0	0.066073	26.0	–0.621116	27.0
云南	–0.459243	22.0	0.121581	21.0	–0.475362	22.0
陕西	–0.397687	19.0	0.134286	19.0	–0.444968	20.0
甘肃	–0.682677	27.0	0.063158	27.0	–0.616788	26.0
青海	–0.925999	30.0	0.000000	30.0	–0.777910	30.0
宁夏	–0.896824	29.0	0.007374	29.0	–0.752243	29.0
新疆	–0.633970	25.0	0.073875	25.0	–0.551969	25.0

4.3 数据压缩方法

4.3.1 主成分分析的基本思想

4.3.1.1 主成分分析的概念

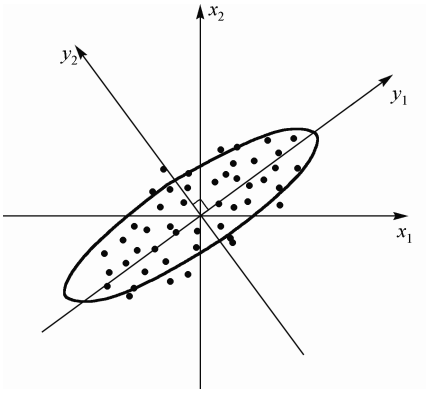
在实际问题中，经常需要研究多元问题，然而在多数情况下，不同变量之间有一定相关性，这必然增加分析问题的复杂性。主成分分析就是通过降维技术把多个指标简化为少数几个综合性指标。例如，在经济管理中用主成分分析法将一些复杂的数据综合成几个商业指数形式，如物价指数、生活费用指数、商业活动指数等。又如，对全国 30 个省、市、自治区的贸易竞争力做综合评价，这时显然需要选取很多指标。如何将这些具有错综复杂关系的指标综合成几个较少的成分，既有利于对问题进行分析和解释，又便于抓住主要矛盾做出科学的评价，这就要用到主成分分析法。

主成分分析法是通过降维技术把多个变量压缩为几个少数主成分的方法，这些主成分保留原始变量的绝大部分信息，它们通常表示为原始变量的线性组合。通过主成分分析，可以从事物之间错综复杂的关系中找出一些主要成分，从而有效利用大量统计数据进行定量分析，揭示变量之间的内在关系，得到对事物特征及其发展规律的一些深层次的启发，把研究工作引向深入。

4.3.1.2 主成分分析的思想

主成分分析的基本思想是设法将众多具有一定相关性的指标，重新组合成一组新的相互无关的综合性指标来代替原来的指标。数学上的处理就是将原来的 p 个指标进行线性组合，将组合的结果作为新的指标。第一个线性组合，即第一个综合性指标，记为 y_1，为了使该线性组合具有唯一性，要求在所有线性组合中 y_1 的方差最大，即 $\mathrm{Var}(y_1)$ 最大，它所包含的信息最多。如果第一个主成分 y_1 不足以代表原来 p 个指标的所有信息，再考虑选取第二个主成分 y_2，并要求 y_1 已有的信息不出现在 y_2 中，即 $\mathrm{Cov}(y_1,y_2)=0$。

右图中变量和成分间的关系：x_1 和 x_2 是沿一定轨迹的分布数据，单独选择 x_1 或 x_2 都会丧失较多的原始信息。作正交(垂直)旋转，得到新的坐标轴 y_1 和 y_2。旋转后数据主要沿 y_1 方向散布，在 y_2 方向的离散程度很低，另外，y_1 和 y_2 是互相垂直的，表明它们互不相关，即使只是单独提取变量 y_1 而放弃变量 y_2，丧失的信息也是微小的。通常把 y_1 称为第 1 主成分，把 y_2 称为第 2 主成分。主成分分析的关键是寻找一组相互正交的向量，原变量乘以该组正交的向量后能得到新变量组。



主成分分析的成分 y_i 和原变量 x_i 之间的关系如下(假定原先有 p 个变量)：

$$\begin{cases} y_1 = u_{11}x_1 + u_{12}x_2 + \cdots + u_{1p}x_p = u_1'x \\ y_2 = u_{21}x_1 + u_{22}x_2 + \cdots + u_{2p}x_p = u_2'x \\ \quad \cdots \\ y_p = u_{p1}x_1 + u_{p2}x_2 + \cdots + u_{pp}x_p = u_p'x \end{cases}$$

式中，u_{ij} 为第 i 个成分 y_i 和第 j 个原变量 x_j 之间的线性相关系数。

y_1，y_2，…，y_p 分别叫作第 1 主成分，第 2 主成分，…，第 p 主成分，其中在选择加权重 u_{i1}，u_{i2}，…，u_{ip} 时，要使 y_1 得到最大解释变异能力，即使 y_1 得到最大变异数，而 y_2 则能对原始资料中尚未被 y_1 解释的变异部分拥有最大解释能力，以此类推，我们可以找出 m 个 y 出来($m \leqslant p$)，通常原始数据有 p 个 x 变量时，经过转换后，仍可找出 p 个 y 出来，不过我们最多选择 m 个 y_i(i=1,2,…,m；$m \leqslant p$)，希望 m 越小越好，但解释能力却能达到 80% 以上。除此之外，m 个 y_i 与原来的 p 个变量 x_j 的最大差别是，原始变量中，多为彼此相关的变量，而经过线性转换后所产生的 m 个 y_i 则为彼此不相关的新变量。

4.3.1.3 主成分的推导

设 $\boldsymbol{y} = a_1x_1 + a_2x_2 + \cdots + a_px_p \equiv \boldsymbol{a}'\boldsymbol{x}$，其中 $\boldsymbol{a} = (a_1, a_2, \cdots, a_p)'$，$\boldsymbol{x} = (x_1, x_2, \cdots, x_p)'$，求主成分就是寻找 $\boldsymbol{x}$ 的线性函数 $\boldsymbol{a}'\boldsymbol{x}$，使相应的方差达到最大，即 $\mathrm{Var}(\boldsymbol{a}'\boldsymbol{x}) = \boldsymbol{a}'\boldsymbol{\Sigma}\boldsymbol{a}$ 达到最大，此处 $\boldsymbol{\Sigma}$ 为 $\boldsymbol{x}$ 的协方差阵。

谱分解定理：设 $\boldsymbol{\Sigma}$ 的特征根为 $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_p > 0$，相应的单位特征向量为 $\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_p$，令 $\boldsymbol{U} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_p)$，则 $\boldsymbol{U}'\boldsymbol{U} = \boldsymbol{U}\boldsymbol{U}' = \boldsymbol{I}$，即 $\boldsymbol{U}$ 为正交阵，$\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_p)$，且 $\boldsymbol{\Sigma} = \boldsymbol{U}\Lambda\boldsymbol{U}'$。

当取 $\boldsymbol{a} = \boldsymbol{u}_1$ 时，$\boldsymbol{u}_1'\boldsymbol{\Sigma}\boldsymbol{u}_1 = \boldsymbol{u}_1'\lambda_1\boldsymbol{u}_1 = \lambda_1$。

于是 $y_1 = \boldsymbol{u}_1'\boldsymbol{x}$ 就是第 1 主成分，它的方差最大，$\mathrm{Var}(y_1) = \mathrm{Var}(\boldsymbol{u}_1'\boldsymbol{x}) = \lambda_1$。

同理，$\mathrm{Var}(y_i) = \mathrm{Var}(\boldsymbol{u}_i'\boldsymbol{x}) = \lambda_i$。另外，

$$\mathrm{Cov}(y_i, y_j) = \mathrm{Cov}(\boldsymbol{u}_i'\boldsymbol{x}, \boldsymbol{u}_j'\boldsymbol{x}) = \boldsymbol{u}_i'\boldsymbol{\Sigma}\boldsymbol{u}_j = \boldsymbol{u}_i'\lambda_i\boldsymbol{u}_j = \boldsymbol{\lambda}_j\boldsymbol{u}_i'\boldsymbol{u}_j = 0 \quad (i \neq j)$$

上述表明：变量 $\boldsymbol{x}$ 的主成分 $\boldsymbol{y}$ 是以 $\boldsymbol{\Sigma}$ 的特征向量为系数的线性组合，它们互不相关，

方差为 $\boldsymbol{\Sigma}$ 的特征根。而 $\boldsymbol{\Sigma}$ 的特征根 $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_p > 0$，所以有 $\mathrm{Var}(y_1) \geqslant \mathrm{Var}(y_2) \geqslant \cdots \geqslant \mathrm{Var}(y_p) > 0$。

定义 $\lambda_k \Big/ \sum_{i=1}^{p} \lambda_i$ 为第 k 个主成分 y_k 的方差贡献率，第 1 个主成分的贡献率最大，表明 y_1 综合原始变量 x_1，x_2，…，x_p 的能力最强，而 y_2，y_3，…，y_p 的综合能力依次递减。若只取 $m(<p)$ 个主成分，则称 $\sum_{i=1}^{m} \lambda_i \Big/ \sum_{i=1}^{p} \lambda_i$ 为主成分 y_1，y_2，…，y_m 的累积方差贡献率，它表示 y_1，y_2，…，y_m 综合 x_1，x_2，…，x_p 的能力，通常所取 m 使得累积方差贡献率不低于 80%即可(也有人认为只要特征根 λ_i 大于 1 即可)。

在实际中，我们通常使用主成分和原变量的相关系数 a_{ij} 来表达它们的关系，而不是直接采用 u_{ij}，即

$$a_{ij} = \rho(x_i, y_j) = \sqrt{\lambda_j} u_{ij} \Big/ \sqrt{\sigma_i} \quad (i, j = 1, 2, \cdots, p)$$

于是 a_{ij} 也称为主成分负荷，矩阵 $\boldsymbol{A} = (a_{ij})$ 称为主成分负荷矩阵。它相当于标准化系数，能反映变量影响大小，如果原始数据已标准化(此时原始数据的标准差 $\sigma_i = 1$)，即基于相关系数矩阵而不是用协差阵计算主成分，则 $a_{ij} = \sqrt{\lambda_j} u_{ij}$。

4.3.2 主成分的基本分析

4.3.2.1 主成分分析步骤

(1)计算主成分对象(pca)

应用 sklearn.decomposition 包的 PCA 函数计算主成分对象。

(2)计算方差贡献率(variances)

每个主成分的贡献率代表原数据总信息量的百分比，其中前 m 个主成分包含的数据信息总量(即其累积方差贡献率)不低于 80%时，可取前 m 个主成分来反映原评价对象。

(3)计算主成分负荷(loadings)

设 Comp_1，Comp_2，…，Comp_m 代表确定的 m 个主成分。

(4)计算主成分得分(scores)

以主成分负荷为权，将各主成分表示为原指标的线性组合，而主成分的含义则由各线性组合中权重较大的指标的综合意义来确定。若取 m=2，则将每个样品的 p 个变量代入公式，即可算出每个样品的主成分得分 scores1 和 scores2，并在平面上作主成分得分的散点图，进而对样品进行分类或对原始数据进行更深入的研究。

(续例 1.3)对例 1.3 的国际贸易竞争力数据应用主成分分析法进行综合评价。以 8 个指标为原始变量，使用 Python，对 30 个地区的竞争力水平做主成分分析，并根据综合得分和综合排名对各地区竞争力水平进行综合评价。

① 计算主成分对象。

In [35]	Z=(MVdata−MVdata.mean())/MVdata.std() from sklearn.decomposition import PCA pca = PCA(n_components=2).fit(Z)

② 确定主成分。

按照累积方差贡献率大于 80%和方差大于 1 的原则，选入两个主成分，其累积方差贡献率为 92.76%，故本例取 n_components=2 是合适的。

In [36]	Vi=pca.explained_variance_;Vi　　#方差 Wi=pca.explained_variance_ratio_;Wi　　#方差贡献率 Wi.sum()　　#累积方差贡献率
Out [36]	array([5.839958 , 1.58060259]) array([0.72999475, 0.19757532]) 0.92757007344990128　　#累积方差贡献率达 92.76%，故选 2 个主成分

③ 主成分负荷。

In [37]	pd.DataFrame(pca.components_.T)　　#主成分负荷

Out [37]

	0	1
0	0.396727	–0.206429
1	0.287392	–0.520928
2	0.307410	–0.481950
3	0.401099	–0.009390
4	0.378909	0.310446
5	0.386403	0.121590
6	0.384584	0.210432
7	0.252681	0.546091

只选择 2 个主成分时，由主成分载荷可以看出，主成分 $Comp_1$ 在【生产总值】、【利用外资】、【进出口额】、【新品出口】和【市场占有】上的载荷值都很大，而 $Comp_2$ 在【从业人员】、【固定资产】和【对外依存】上有较大的载荷。

① 主成分得分。

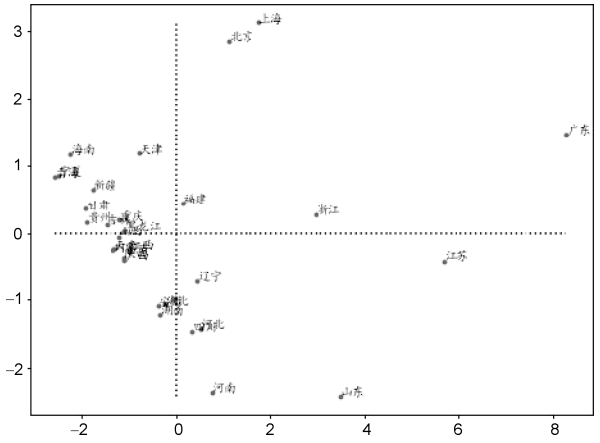
In [38]	Si=pca.fit_transform(Z);Si SR=pd.DataFrame(Si,columns=['Comp1','Comp2'],index=MVdata.index);SR

Out [38]

地区	Comp1	Comp2
北京	1.105610	2.858102
天津	–0.786233	1.184035
河北	0.529178	–1.429212
山西	–1.217342	–0.052739
内蒙古	–1.339672	–0.254383
辽宁	0.449117	–0.710646
……		

由加权法估计出综合得分，以各主成分的方差贡献占两个主成分总方差贡献的比重作为权重进行加权汇总，得出各省、市、自治区的综合得分。

结合各省、市、自治区在主成分上的得分和综合得分，就可以对各省、市、自治区的国际竞争力进行评价了。以第一主成分为横轴，第二主成分为纵轴，绘制各省、市、自治区的成分图，见下页图。

In [39]	plt.plot(SR.Comp1,SR.Comp2,'.'); dm.hvline(SR.Comp1,SR.Comp2,SR.index);
Out [39]	

4.3.2.2 主成分综合评价

最后，以各主成分的方差为权，将各主成分加权并得到综合得分。

$$\text{Comp}=\frac{w_1\text{Comp}_1+w_2\text{Comp}_2+\cdots+w_m\text{Comp}_m}{w_1+w_2+\cdots+w_m}=\sum_{j=1}^{m}w_j\text{Comp}_j$$

式中，w_j 是主成分的权重，利用总得分就可以得到得分名次。

In [40]	SR['Comp']=Si.dot(Wi);SR SR['Rank']=SR.Comp.rank(ascending=False);SR
Out [40]	

地区	Comp1	Comp2	Comp	Rank
北京	1.105610	2.858102	1.371780	6.0
天津	−0.786233	1.184035	−0.340010	12.0
河北	0.529178	−1.429212	0.103920	9.0
山西	−1.217342	−0.052739	−0.899073	21.0
……				
甘肃	−1.916851	0.363137	−1.327544	26.0
青海	−2.553884	0.835643	−1.699220	30.0
宁夏	−2.478707	0.856885	−1.640144	29.0
新疆	−1.779668	0.645596	−1.171594	25.0

不过，Python 函数本身并没有给出进行综合得分和排名的功能，所以我们自定义了一个进行主成分综合分析的函数 PCrank()，方便大家使用。

In [41]	da.PCrank(MVdata,m=2)　　#自定义主成分综合评价函数

Out [41] 方差贡献:

	Variances	Explained	Cumulative
Comp1	5.8400	72.9995	72.9995
Comp2	1.5806	19.7575	92.7570
Comp3	0.3338	4.1725	96.9295
Comp4	0.1578	1.9722	98.9017

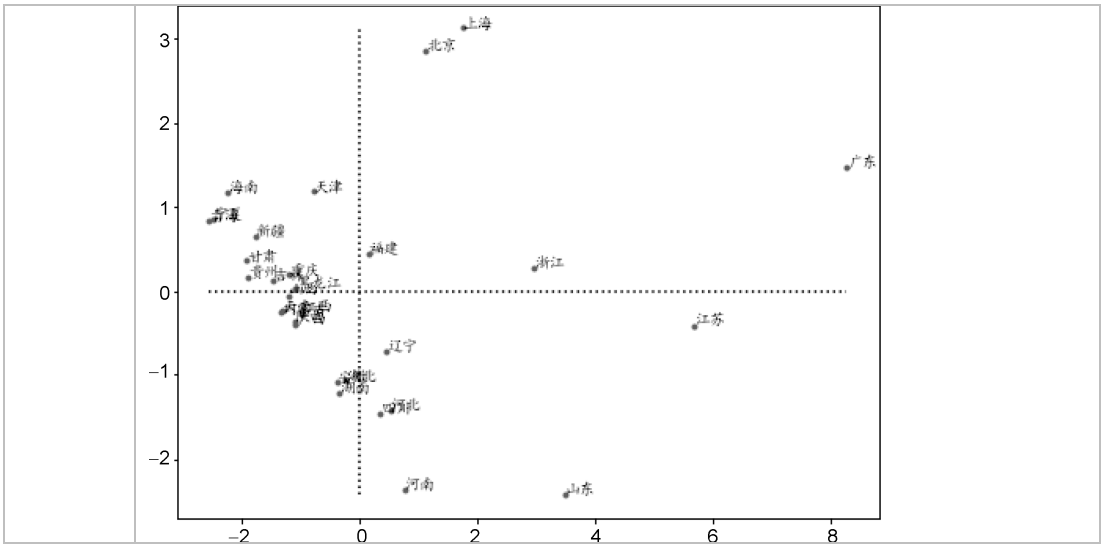
Comp5	0.0555	0.6938	99.5956
Comp6	0.0167	0.2093	99.8049
Comp7	0.0086	0.1077	99.9125
Comp8	0.0070	0.0875	100.0000

主成分负荷:

	Comp1	Comp2
生产总值	0.3967	–0.2064
从业人员	0.2874	–0.5209
固定资产	0.3074	–0.4819
利用外资	0.4011	–0.0094
进出口额	0.3789	0.3104
新品出口	0.3864	0.1216
市场占有	0.3846	0.2104
对外依存	0.2527	0.5461

综合得分与排名:

	Comp1	Comp2	Comp	Rank
地区				
北京	1.1056	2.8581	1.3718	6.0
天津	–0.7862	1.1840	–0.3400	12.0
河北	0.5292	–1.4292	0.1039	9.0
山西	–1.2173	–0.0527	–0.8991	21.0
内蒙古	–1.3397	–0.2544	–1.0282	23.0
辽宁	0.4491	–0.7106	0.1874	8.0
吉林	–1.4755	0.1245	–1.0525	24.0
黑龙江	–1.1038	0.0286	–0.8001	17.0
上海	1.7450	3.1264	1.8916	5.0
江苏	5.6754	–0.4195	4.0601	2.0
浙江	2.9599	0.2729	2.2146	3.0
安徽	–0.3869	–1.0965	–0.4991	15.0
福建	0.1598	0.4401	0.2036	7.0
江西	–0.9970	–0.2646	–0.7801	16.0
山东	3.4916	–2.4144	2.0718	4.0
河南	0.7730	–2.3590	0.0982	10.0
湖北	–0.2515	–1.0713	–0.3953	13.0
湖南	–0.3440	–1.2216	–0.4925	14.0
广东	8.2677	1.4782	6.3274	1.0
广西	–1.0983	–0.3994	–0.8807	20.0
海南	–2.2387	1.1787	–1.4013	28.0
重庆	–1.2093	0.1953	–0.8442	18.0
四川	0.3418	–1.4645	–0.0398	11.0
贵州	–1.8899	0.1649	–1.3470	27.0
云南	–1.3237	–0.2349	–1.0127	22.0
陕西	–1.1071	–0.3602	–0.8794	19.0
甘肃	–1.9169	0.3631	–1.3275	26.0
青海	–2.5539	0.8356	–1.6992	30.0
宁夏	–2.4787	0.8569	–1.6401	29.0
新疆	–1.7797	0.6456	–1.1716	25.0



就综合得分来看，广东、江苏、山东、浙江和上海 5 省的得分位列前五位，海南、宁夏和青海得分位列后三位。

从现实经济发展和地理位置来看，东部沿海地区的贸易国际竞争力明显优于中西部地区，说明地理位置对国际贸易的发展发挥着重要的作用。总体来看，经济发展水平较高的省、市、自治区，其城镇贸易国际竞争力也相对较高；经济较落后的地区，其贸易国际竞争力也相对较低。

4.4 聚类分析方法

4.4.1 聚类分析的概念

4.4.1.1 聚类分析的起源

聚类分析(cluster analysis)是研究“物以类聚”的一种现代统计分析方法，如不同地区城镇居民收入和消费状况的分类研究、区域经济及社会发展水平的分析，以及全国区域经济综合区划。过去人们受分析工具的限制，主要依靠经验和专业知识做定性分类处理，很少利用统计方法，致使许多分类带有主观性和随意性，不能很好地揭示客观事物内在的本质差别和联系，特别是对于多个指标的分类问题，定性分类更难以实现准确分类。为了克服定性分类的不足，多元统计分析中引入数值分类方法，形成了聚类分析分支。

近年来聚类分析发展很快，在社会、经济、管理、地质勘探、天气预报、生物分类、考古、医学、心理学以及制定国家标准和区域标准等许多方面的应用都很有成效，因而也成为目前较为流行的多元统计分析方法之一。例如，在古生物研究中，通过挖掘出来的一些骨骼的形状和大小将它们科学地进行分类。在地质勘探中，通过矿石标本的物探、

化探指标将标本进行分类。在经济区域的划分中，根据各主要经济指标将全国各省、市、自治区分成几个区域。

4.4.1.2 聚类思想与类型

聚类分析的基本思路是把分类对象按一定规则分成若干类，这些类不是事先给定的，而是根据数据的特征来确定的。在同一类中，这些对象在某种意义上趋向于彼此相似；而在不同类中，对象趋向于不相似。

在聚类分析中，基本思想是，认为所研究的样品或变量之间存在着程度不同的相似性(亲疏关系)。根据一批样品的多个观测变量，具体找出一些能够度量样品(或变量)之间相似程度的统计量，以这些统计量为划分类型的依据，把一些相似程度较大的样品(或变量)聚为一类，把另外一些彼此之间相似程度较大的样品(或变量)又聚为另一类，关系密切的聚到一个小的分类中，关系疏远的聚到一个大的分类中，直到把所有样品(或变量)都聚类完毕，把不同的类型一一划分出来，形成一个由小到大的分类系统。最后再把整个分类系统画成一张聚类图，用它把所有样品(或变量)间的亲疏关系表示出来。

常见的聚类分析方法有系统聚类法、快速聚类法、有序聚类法和模糊聚类法等，本书重点介绍目前常用的系统聚类法，其他方法参考有关书籍。

通常根据分类对象的不同分为两类：一类是对样品进行分类处理，叫 Q 型聚类；另一类是对变量进行分类处理，叫 R 型聚类。

4.4.1.3 聚类统计量

聚类分析的基本原则是将有较大相似性的对象归为同一类，而将差异较大的个体归入不同的类。为了将样品聚类，就需要研究样品之间的关系。一种方法是将每个样品看作 p 维空间的一个点，并在空间定义距离，距离较近的点归为一类，距离较远的点归入不同的类。对变量通常计算它们的相关系数，性质越接近的变量，其相关系数越接近 1(或 –1)，彼此越无关的变量，其相关系数越接近 0。比较相近的变量归为一类，不怎么相近的变量归入不同的类。

可进行聚类的统计量有距离和相似系数两种：

$$\text{聚类统计量}\begin{cases}\text{距离（样品)}\\ \text{相关系数（变量）}\end{cases}$$

对样品进行聚类时，我们把样品间的“靠近”程度用某种距离来刻画；对指标的聚类，往往用某种相关系数来刻画。

(1)距离的计算

当选用 n 个样品、p 个指标时，就可以得到一个 $n\times p$ 的数据矩阵 $\boldsymbol{X}=(x_{ij})_{n\times p}$。该矩阵的元素 x_{ij} 表示第 i 个样品的第 j 个变量值。

为了直观显示样品之间的距离，举一个两变量在平面上的例子。

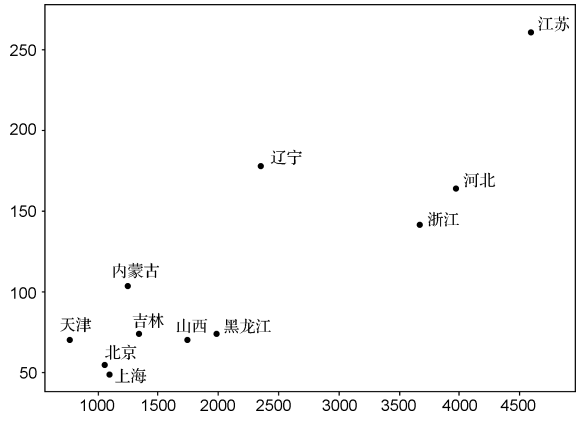
从前面的竞争力数据中取出任意两个变量，在直角坐标系中显示它们在空间的距离分布情况，如取变量 X1(从业人员)和 X2(固定资产)的前 11 个样品(地区)。

In [42]

```
X12=YXdata[['X1','X2']][:11];X12  #取变量 X1 和 X2 的前 11 个数据
plt.plot(X12.X1,X12.X2,'.')
for i in range(11):  plt.text(X12.X1[i],X12.X2[i],X12.index[i])
```

Out [42]

地区	X1	X2
北京	1069.70	55.7893
天津	763.16	70.6767
河北	3962.42	163.8933
山西	1738.90	70.7306
内蒙古	1249.30	103.6517
辽宁	2364.90	177.2629
吉林	1337.80	74.4171
黑龙江	1977.80	74.7538
上海	1104.33	49.6207
江苏	4758.23	266.9262
浙江	3680.00	141.8528



由于只有两个变量，所以从散点图上就可以直观地将这些地区样品分为几类，但当变量多于两个时，这种方法显然是不行的。下面给出计算距离的常用方法。

设 $x_{ij}(i=1,2,\cdots,n;\ j=1,2,\cdots,p)$ 为第 i 个样品的第 j 个指标的观测数据，即每个样品有 p 个变量，则每个样品都可以看成 p 维空间中的一个点，n 个样品就是 p 维空间中的 n 个点，定义 d_{ij} 为样品 x_i 与 x_j 的距离，于是得到 $n\times n$ 的距离矩阵：

$$\boldsymbol{D}=(d_{ij})_{n\times n}=\begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ d_{n1} & d_{n2} & \cdots & d_{nn} \end{bmatrix}$$

为了计算平面上各点之间的距离 d_{ij}，在聚类分析中对连续变量常用的距离有以下几种。

① 欧氏距离(Euclidean)，通常是一种数学意义上的距离。

$$d_{ij}(2)=\left[\sum_{k=1}^{p}(x_{ik}-x_{jk})^2\right]^{\frac{1}{2}}$$

② 马氏距离(Mahalanobis)，是一种统计意义上的距离，可看作欧氏距离的推广。

$$d_{ij}(M)=(\boldsymbol{x}_i-\boldsymbol{x}_j)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i-\boldsymbol{x}_j)$$

式中，$\boldsymbol{x}_i$ 为样品 i 的 p 个指标组成的行向量，$\boldsymbol{\Sigma}$ 为协方差阵。

相对于欧氏距离，马氏距离有其优点：马氏距离既排除了各指标间的相关性干扰，又消除了各指标的量纲。下面是欧氏距离算出的距离相似矩阵(Python 默认为欧氏距离)。

In [43]	Z12=(X12−X12.mean())/X12.std()　　#数据标准化 import scipy.cluster.hierarchy as sch　　#加载系统聚类包 D12=sch.distance.pdist(Z12);D12　　#样品间距离
Out [43]	array([0.317,　2.675,　0.542,　0.725,　2.046,　0.341,　0.728,　0.095, 4.16 ,　2.315,　2.741,　0.721,　0.608,　1.979,　0.428,　0.9 , 0.402,　4.153,　2.401,　2.15 ,　2.196,　1.197,　2.352,　1.977, 2.711,　1.642,　0.389,　0.609,　1.652,　0.301,　0.186,　0.564, 3.675,　1.783,　1.371,　0.44 ,　0.689,　0.811,　3.553,　1.884, 1.708,　1.552,　2.116,　2.215,　1.105,　0.473,　0.407,　3.82 , 2.001,　0.746,　3.521,　1.606,　4.213,　2.346,　2.025])

这是下面进行 Q 型聚类分析的出发点。首先将距离最小的两个样品聚为一类，例如，前面的第 1 个样品和第 9 个样品聚为一类（d=0.095）。

在实际聚类分析中，很多情况下都是对样品做聚类，所以下面重点介绍针对样品的 Q 型聚类方法。

由于上述距离是样品在平面坐标上的两两之间的关系，并不能反映它们的多维空间的关系，需要进一步进行聚类分析。

(2) 相关系数

常用相关系数矩阵 $\boldsymbol{R}=(r_{ij})_{n\times p}$ 表示变量间的相关性。

$$\boldsymbol{R}=\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1p} \\ r_{21} & r_{22} & \cdots & r_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ r_{p1} & r_{p2} & \cdots & r_{pp} \end{bmatrix}=\begin{bmatrix} 1 & r_{12} & \cdots & r_{1p} \\ r_{21} & 1 & \cdots & r_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ r_{p1} & r_{p2} & \cdots & 1 \end{bmatrix}$$

式中，r_{ij} 为任意两变量之间的简单相关系数。

4.4.2　系统聚类方法

4.4.2.1　系统聚类的基本思想

确定了距离后就要进行分类，分类有许多种方法，最常用的一类方法是在样品距离的基础上定义类与类之间的距离，首先将 n 个样品分成 n 类，每个样品自成一类，然后每次将具有最小距离的两类合并，合并后重新计算类与类之间的距离，这个过程一直继续到所有的样品归为一类为止，并把这个过程作成一张聚类图，由聚类图可方便地进行分类。因为聚类图类似于一张系统图，所以这类方法就称为系统聚类法（hierachical clustering method）。系统聚类的方法是目前在实际中使用最多的一类方法。从上面的分析可以看出，虽然我们已定义样品之间的距离，但在实际计算过程中还要定义类与类之间的距离，如何定义类与类之间的距离，也有许多种方法，不同的定义方法就产生了不同的系统聚类方法，常用的有以下六种。

① 最短距离法：类与类之间的距离等于两类最近样品之间的距离。

② 最长距离法：类与类之间的距离等于两类最远样品之间的距离。

③ 中间距离法：最长距离法夸大了类间距离，最短距离法低估了类间距离，介于两者间的距离法即中间距离法。

④ 类平均法：类与类之间的距离等于各类元素两两之间的平方距离的平均值。

⑤ 重心法：类与类之间的距离定义为对应这两类重心(均值)之间的距离。

⑥ 离差平方和法(Ward 法)：基于方差分析的思想，如果类分得正确，同类样品之间的离差平方和应当较小，类与类之间的离差平方和应当较大。

这六种系统聚类法的并类原则和过程完全相同，不同之处在于类与类之间的距离定义不同。

4.4.2.2 系统聚类的基本步骤

系统聚类的基本步骤如下：

① 计算 n 个样品两两间的距离阵，记作 $\boldsymbol{D}=\{d_{ij}\}_{n\times n}$；

② 构造 n 个类，每个类只包含一个样品；

③ 合并距离最近的两类为一个新类；

④ 计算新类与当前各类的距离，若类个数为 1，则转到步骤(5)，否则回到步骤(3)；

⑤ 绘制系统聚类图；

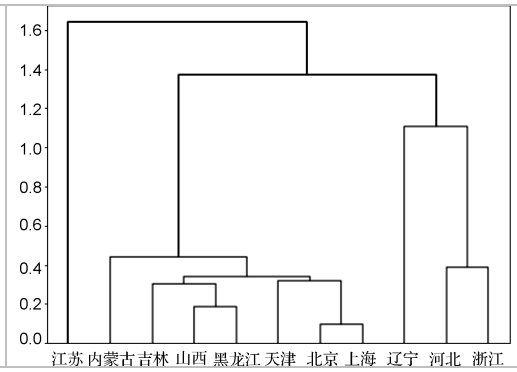
⑥ 根据系统聚类图确定类的个数和类的内容。

下面应用前面的数据框 X12 进行系统聚类。距离采用欧氏距离(Python 默认)，方法使用最长距离法(Python 默认)。开始有 11 类，即每个样品自成一类，这 11 类之间的距离就等于 11 个样品(地区)之间的距离，距离矩阵记为 $\boldsymbol{D}$，其最小元素是 $\boldsymbol{D}(0,8)=0.095$，故第一步就可将类 0(北京)和类 8(上海)合并成一个新类，以此类推，然后计算新类与其他类之间的距离。

首先使用默认的最长距离法进行聚类，具体如下。

In [44]	H1=sch.linkage(D12);H1 #系统聚类过程，默认方法 method='complete'
Out [44]	array([[0. , 8. , 0.09530393, 2.], [3. , 7. , 0.18639825, 2.], [6. , 12. , 0.30140255, 3.], [1. , 11. , 0.31684622, 3.], [13. , 14. , 0.34073179, 6.], [2. , 10. , 0.38877071, 2.], [4. , 15. , 0.43997321, 7.], [5. , 16. , 1.10541379, 3.], [17. , 18. , 1.37100242, 10.], [9. , 19. , 1.64227407, 11.]])
In [45]	sch.dendrogram(H1,labels=X12.index); #系统聚类图

Out [45]



从聚类分析结果可以看到，如果聚为三类，则第一类包括江苏，第二类包括辽宁、河北和浙江，第三类包括内蒙古、吉林、山西、黑龙江、天津、北京和上海。

In [46]

```
pd.DataFrame(sch.cut_tree(H1),index=X12.index)          #聚类划分
```

Out [46]

地区	0	1	2	3	4	5	6	7	8	9	10
北京	0	0	0	0	0	0	0	0	0	0	0
天津	1	1	1	1	0	0	0	0	0	0	0
河北	2	2	2	2	1	1	1	1	1	0	0
山西	3	3	3	3	2	0	0	0	0	0	0
内蒙古	4	4	4	4	3	2	2	0	0	0	0
辽宁	5	5	5	5	4	3	3	2	1	0	0
吉林	6	6	6	3	2	0	0	0	0	0	0
黑龙江	7	7	3	3	2	0	0	0	0	0	0
上海	8	0	0	0	0	0	0	0	0	0	0
江苏	9	8	7	6	5	4	4	3	2	1	0
浙江	10	9	8	7	6	5	1	1	1	0	0

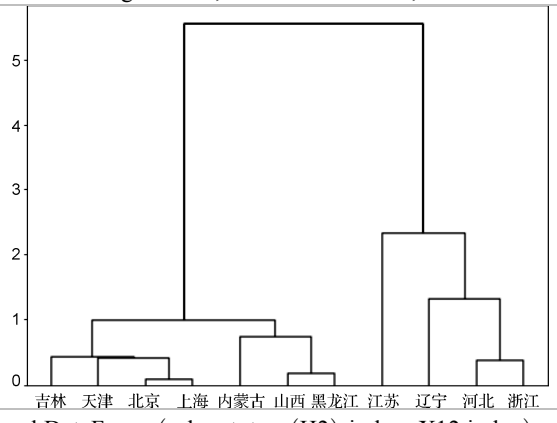
从上表可以看出，第 1 次北京和上海最为接近，聚为一类；第 2 次山西和黑龙江聚为一类，以此类推。

下面使用 ward 方法进行聚类，通常效果比较好。

In [47]

```
H2=sch.linkage(D12,method='ward');H2                    #系统聚类过程，方法='ward'
sch.dendrogram(H2,labels=X12.index);
```

Out [47]



In [48]

```
pd.DataFrame(sch.cut_tree(H2),index=X12.index)          #聚类划分
```

Out [48]

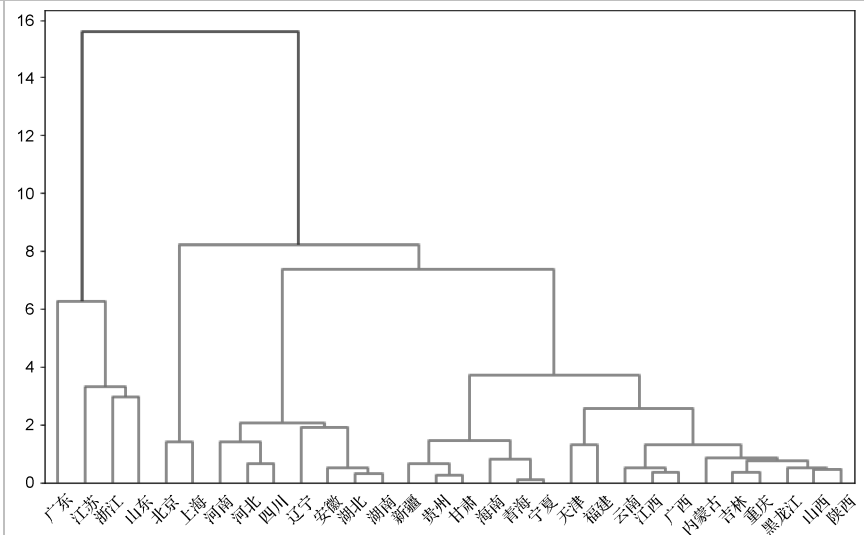
	0	1	2	3	4	5	6	7	8	9	10
地区											
北京	0	0	0	0	0	0	0	0	0	0	0
天津	1	1	1	1	0	0	0	0	0	0	0
河北	2	2	2	2	1	1	1	1	1	1	0
山西	3	3	3	3	2	2	2	0	0	0	0
内蒙古	4	4	4	4	3	3	2	0	0	0	0
辽宁	5	5	5	5	4	4	3	2	1	1	0
吉林	6	6	6	6	5	0	0	0	0	0	0
黑龙江	7	7	3	3	2	2	2	0	0	0	0
上海	8	0	0	0	0	0	0	0	0	0	0
江苏	9	8	7	7	6	5	4	3	2	1	0
浙江	10	9	8	2	1	1	1	1	1	1	0

继续对我国 30 个省、市、自治区 2011 年的对外贸易数据进行 8 个变量的样品聚类，根据聚类结果做国际竞争力划分。虽然 Python 要通过编程来进行统计分析，使得许多人望而却步，实际上，如果使用熟练，用 Python 进行分析还是非常灵活的。下面用一步法对地区国际竞争力进行聚类分析。由于经济管理数据通常变化较大，故需先对其进行标准化再聚类。

In [49]

```
Z=(MVdata-MVdata.mean())/MVdata.std()
D=sch.distance.pdist(Z);
H=sch.linkage(D,method='ward');
sch.dendrogram(H,labels=MVdata.index);
```

Out [49]



In [50]

```
pd.DataFrame(sch.cut_tree(H),index=MVdata.index).iloc[:,-5:]   #聚为 5 到 1 类
```

Out [50]

	25	26	27	28	29
地区					
北京	0	0	0	0	0
天津	1	1	1	0	0
河北	2	2	1	0	0

	山西	1	1	1	0	0
	内蒙古	1	1	1	0	0
	辽宁	2	2	1	0	0
	吉林	1	1	1	0	0
	黑龙江	1	1	1	0	0
	上海	0	0	0	0	0
	江苏	3	3	2	1	0
	浙江	3	3	2	1	0
	安徽	2	2	1	0	0
	福建	1	1	1	0	0
	江西	1	1	1	0	0
	山东	3	3	2	1	0
	河南	2	2	1	0	0
	湖北	2	2	1	0	0
	湖南	2	2	1	0	0
	广东	4	3	2	1	0
	广西	1	1	1	0	0
	海南	1	1	1	0	0
	重庆	1	1	1	0	0
	四川	2	2	1	0	0
	贵州	1	1	1	0	0
	云南	1	1	1	0	0
	陕西	1	1	1	0	0
	甘肃	1	1	1	0	0
	青海	1	1	1	0	0
	宁夏	1	1	1	0	0
	新疆	1	1	1	0	0

综合考虑以上分析结果，我们建立全国 30 个地区的外贸国际竞争力的分类情况，如表 4-3 所示。

表 4-3　按类整理聚类图结果

分　　类	类(1)	类(2)		
分两类	广东、江苏、浙江、山东	北京、天津、河北、山西、内蒙古、辽宁、吉林、黑龙江、上海、安徽、福建、江西、河南、湖北、湖南、广西、海南、重庆、四川、贵州、云南、陕西、甘肃、青海、宁夏、新疆		
分三类	类(1)	类(2)	类(3)	
	广东、江苏、浙江、山东	北京、上海	天津、河北、山西、内蒙古、辽宁、吉林、黑龙江、安徽、福建、江西、河南、湖北、湖南、广西、海南、重庆、四川、贵州、云南、陕西、甘肃、青海、宁夏、新疆、	
分四类	类(1)	类(2)	类(3)	类(4)
	广东、江苏、浙江、山东	北京、上海	河南、河北、四川、辽宁、安徽、湖北、湖南	天津、山西、内蒙古、吉林、黑龙江、福建、江西、广西、海南、重庆、四川、贵州、云南、陕西、甘肃、青海、宁夏、新疆

从表 4-3 可以看出，江苏、浙江、山东、广东的国际贸易竞争力与其他省、市、自治

区有较显著的差异，这是符合实际情况的，于是可以将我国国际贸易竞争力水平大致分类如下。

如果按高竞争力和低竞争力进行分类，则分两类：江苏、浙江、山东、广东为高竞争力地区，其他为低竞争力地区。

如果按高竞争力、中等竞争力和低竞争力进行分类，则分三类：江苏、浙江、山东、广东为高竞争力地区，北京、上海为中等竞争力地区，其他为低竞争力地区。

如果按高竞争力、中等偏上竞争力、中等偏下竞争力和低竞争力进行分类，则分四类：

江苏、浙江、山东、广东为高竞争力地区，北京、上海为中等偏上竞争力地区，河南、河北、四川、辽宁、安徽、湖北、湖南为中等偏下竞争力地区，天津、山西、内蒙古、吉林、黑龙江、福建、江西、广西、海南、重庆、四川、贵州、云南、陕西、甘肃、青海、宁夏、新疆为低竞争力地区。

数据与练习 4

4.1 经济数据：收集 2000－2010 年共 12 年财政收入相关数据，分别是财政收入(y，百亿元)、国民生产总值(x1，百亿元)、税收(x2，百亿元)、进出口贸易总额(x3，百亿元)、经济活动人口(x4，百万人)。

year	y	x1	x2	x3	x4
2000	29.37	185.98	28.22	55.60	653.23
2001	31.49	216.63	29.90	72.26	660.91
2002	34.83	266.52	32.97	91.20	667.82
2003	43.49	345.61	42.55	112.71	674.68
2004	52.18	466.70	51.27	203.82	681.35
2005	62.42	574.95	60.38	235.00	688.55
2006	74.08	668.51	69.10	241.34	697.65
2007	86.51	731.43	82.34	269.67	708.00
2008	98.76	769.67	92.63	268.58	720.87
2009	114.44	805.79	106.83	298.96	727.91
2010	133.95	882.28	125.82	392.74	739.92
2011	163.86	943.46	153.01	421.93	744.32

(1) 试将这组数据输入电子表格。

(2) 分别用 Python 的 read_csv() 和 read_excel() 函数读取。

(3) 试用 Python 函数获取 2006—2011 年的数据，以及 2006—2011 年的国民生产总值和经济活动人口数据。

(4) 进行多元相关分析。

(5) 进行多元回归分析。

4.2 为了研究 31 个省、市、自治区 2007 年城镇居民生活消费的分布规律，根据调

查资料做区域消费类型划分，指标名称如下。此例样品数 n=31，变量个数 p=8。数据来源于《2008 中国统计年鉴》。

食品：人均食品支出(元/人)；

衣着：人均衣着商品支出(元/人)；

设备：人均家庭设备用品及服务支出(元/人)；

医疗：人均医疗保健支出(元/人)；

交通：人均交通和通信支出(元/人)；

教育：人均娱乐教育文化服务支出(元/人)；

居住：人均居住支出(元/人)；

杂项：人均杂项商品和服务支出(元/人)。

地区	食品	衣着	设备	医疗	交通	教育	居住	杂项
北京	4934.05	1512.88	981.13	1294.07	2328.51	2383.96	1246.19	649.66
天津	4249.31	1024.15	760.56	1163.98	1309.94	1639.83	1417.45	463.64
河北	2789.85	975.94	546.75	833.51	1010.51	895.06	917.19	266.16
山西	2600.37	1064.61	477.74	640.22	1027.99	1054.05	991.77	245.07
内蒙古	2824.89	1396.86	561.71	719.13	1123.82	1245.09	941.79	468.17
辽宁	3560.21	1017.65	439.28	879.08	1033.36	1052.94	1047.04	400.16
吉林	2842.68	1127.09	407.35	854.8	873.88	997.75	1062.46	394.29
黑龙江	2633.18	1021.45	355.67	729.55	746.03	938.21	784.51	310.67
上海	6125.45	1330.05	959.49	857.11	3153.72	2653.67	1412.1	763.80
江苏	3928.71	990.03	707.31	689.37	1303.02	1699.26	1020.09	377.37
浙江	4892.58	1406.2	666.02	859.06	2473.4	2158.32	1168.08	467.52
安徽	3384.38	906.47	465.68	554.44	891.38	1169.99	850.24	309.3
福建	4296.22	940.72	645.4	502.41	1606.9	1426.34	1261.18	375.98
江西	3192.61	915.09	587.4	385.91	732.97	973.38	728.76	294.60
山东	3180.64	1238.34	661.03	708.58	1333.63	1191.18	1027.58	325.64
河南	2707.44	1053.13	549.14	626.55	858.33	936.55	795.39	300.19
湖北	3455.98	1046.62	550.16	525.32	903.02	1120.29	856.97	242.82
湖南	3243.88	1017.59	603.18	668.53	986.89	1285.24	869.59	315.82
广东	5056.68	814.57	853.18	752.52	2966.08	1994.86	1444.91	454.09
广西	3398.09	656.69	491.03	542.07	932.87	1050.04	803.04	277.43
海南	3546.67	452.85	519.99	503.78	1401.89	837.83	819.02	210.85
重庆	3674.28	1171.15	706.77	749.51	1118.79	1237.35	968.45	264.01
四川	3580.14	949.74	562.02	511.78	1074.91	1031.81	690.27	291.32
贵州	3122.46	910.3	463.56	354.52	895.04	1035.96	718.65	258.21
云南	3562.33	859.65	280.62	631.70	1034.71	705.51	673.07	174.23
西藏	3836.51	880.1	271.29	272.81	866.33	441.02	628.35	335.66
陕西	3063.69	910.29	513.08	678.38	866.76	1230.74	831.27	332.84
甘肃	2824.42	939.89	505.16	564.25	861.47	1058.66	768.28	353.65
青海	2803.45	898.54	484.71	613.24	785.27	953.87	641.93	331.38
宁夏	2760.74	994.47	480.84	645.98	859.04	863.36	910.68	302.17
新疆	2760.69	1183.69	475.23	598.78	890.30	896.79	736.99	331.80

(1)用综合评价法进行综合分析。

(2)用主成分分析法进行综合分析。

(3)用系统聚类分析法进行聚类。

4.3 USJudgeRatings 数据集(来自 R 语言 datasets 包)包含了律师对美国高等法院法官的评分。数据框包含 43 个观测和 12 个变量(我们只需后 11 个变量)。CONT 为律师与法官的接触次数，INTG 为(法官的)正直程度，DMNR 为风度，DILG 为勤勉度，CFMG 为案例流程管理水平，DECI 为决策效率，PREP 为审理前的准备工作，FAMI 为对法律的熟悉程度，ORAL 为口头裁决的可靠度，WRIT 为书面裁决的可靠度，PHYS 为体能，RTEN 为是否值得保留。

(1)用综合评价法进行综合分析。

(2)用主成分分析法进行综合分析。

(3)用系统聚类分析法进行聚类。

4.4 UScereal 数据集(来自 R 语言 MASS 包)给出谷物营养的数据。变量 mfr 为生产厂家名，calories、protein、fat、sodium、fibre、carbo、sugars 均为谷物的营养成分。

(1)用综合评价法进行综合分析。

(2)用主成分分析法进行综合分析。

(3)用系统聚类分析法进行聚类。

第 5 章　时序数据的模型分析

5.1　时间序列简介

5.1.1　时间序列的概念

(1)定义

时间序列指将同一统计指标的数值按其发生的时间先后顺序排列而成的数列。时间序列分析的主要目的是根据已有的历史数据对未来进行预测。

(2)要素

时间序列由两个基本要素组成：一个是资料所属的时间；另一个是时间上的统计指标数值。

(3)作用

① 时间序列可以描述社会经济现象在不同时间的发展状态和过程。

② 时间序列可以研究社会经济现象的发展趋势和速度，以及掌握发展变化的规律。

③ 借助时间序列可以进行分析和预测。

(4)分析

时间序列分析是根据系统观测得到的时间序列数据，通过曲线拟合和参数估计(如非线性最小二乘法)来建立数学模型的理论和方法。时间序列分析常用在国民经济宏观控制、区域综合发展规划、企业经营管理、市场潜量预测、气象预报、水文预报、地震前兆预报、农作物病虫灾害预报、环境污染控制、生态平衡、天文学和海洋学等方面。

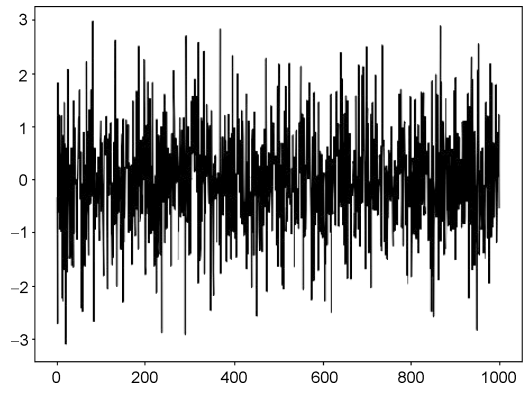
5.1.2　时间序列的模拟

5.1.2.1　平稳序列模拟

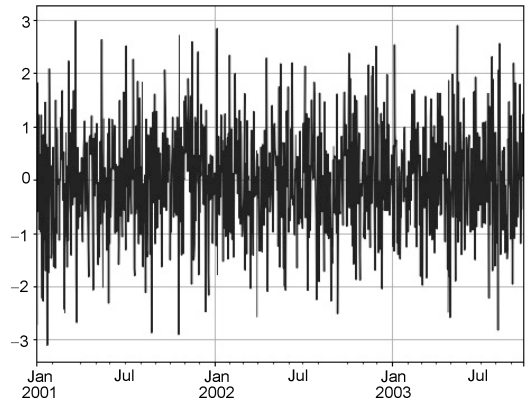
设 $R \sim N(\mu,\sigma^2)$，令 $\mu = 0, \sigma=1$，以下代码将产生 1000 个平稳随机过程序列。

(1)随机游走序列

In　[1]	n=1000 rd=np.random.randn(n) plt.plot(rd);

Out [1]	

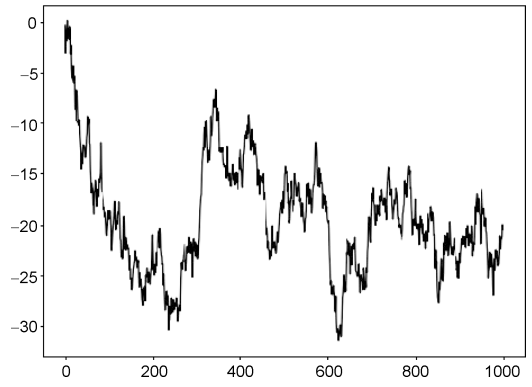
(2) 平稳时间序列

In [2]	rd_ts=pd.Series(rd,index=pd.period_range('2001-01-01',periods=n)) rd_ts.plot(grid=True)
Out [2]	

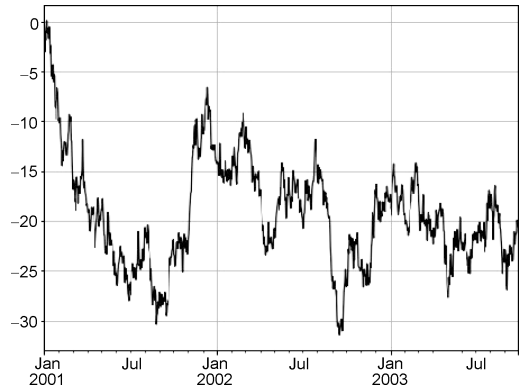
5.1.2.2 非平稳序列模拟

设 $R \sim N(\mu,\sigma^2)$，令 $\mu = 0$, σ=1，而一个累积正态分布随机变量就是一个非平稳的时间序列，以下代码将产生 1000 个非平稳随机过程序列。

(1) 布朗运动序列

In [3]	plt.plot(rd.cumsum())
Out [3]	

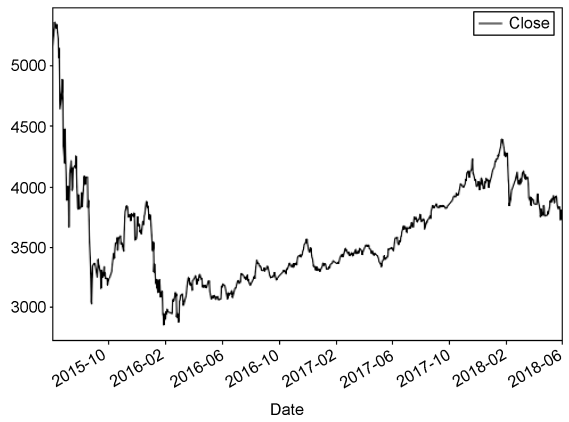
(2)非平稳时间序列

In [4]	rd_ts.cumsum().plot(grid=True)
Out [4]	

5.1.3 时间序列的读取

5.1.3.1 股票指数数据的读取

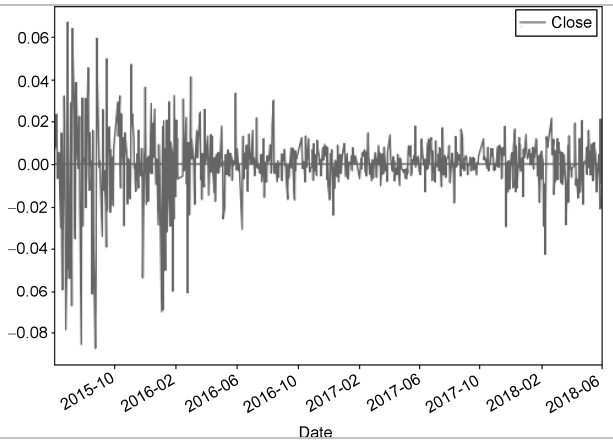
在例 1.3 中，我们收集了 2015 年 6 月 3 日至 2018 年 6 月 1 日沪深 300 指数的收盘价(Close)数据，共 732 个，存放在 PyDm_data.xlsx 文档的股票数据【TSdata】表中。

In [5]	TSdata=pd.read_excel('PyDm_data.xlsx','TSdata',index_col=0); TSdata.head()
Out [5]	Close Date 2015-06-03 5143.590 2015-06-04 5181.416 2015-06-05 5230.552 2015-06-08 5353.751 2015-06-09 5317.461 ……
In [6]	TSdata.plot();
Out [6]	

显然，股票收盘价指数数据是典型的时间序列数据。

5.1.3.2 股票收益率的计算

计算方法及其 Python 实现如下。

In [7]	def Return(Yt):　　#计算收益率 　　Rt=Yt/Yt.shift(1)−1　　#Yt.diff()/Yt.shift(1)，shift(1)表示滞后一阶 　　return(Rt) Rt=Return(TSdata)
Out [7]	Close Date 2015-06-03　NaN 2015-06-04　0.0074 2015-06-05　0.0095 2015-06-08　0.0236 2015-06-09 -0.0068 ……
In [8]	Rt.plot().axhline(y=0)
Out [8]	

可以看到，股票收益率是围绕 0 上下波动的时间序列数据。

5.2 时间序列分析模型

时间序列分析模型最著名的当属博克思(Box)和詹金斯(Jenkins)于 20 世纪 70 年代初提出的时间序列预测模型 ARIMA，又称为博克思-詹金斯模型(Box-Jenkins 模型)，亦称 B-J 方法，全称为自回归积分移动平均模型(AutoRegressive Integrated Moving Average model，ARIMA)。它是一种精度较高的时间序列短期分析方法，其基本思想是，某些时间序列是依赖于时间 t 的一组随机变量，构成该时间序列的单个序列值虽然具有不确定性，但整个序列的变化却有一定的规律性，可以用相应的数学模型近似描述，通过对该数学模型的分析研究，能够从本质上认识时间序列的结构与特征，并得到最有效的预测结果。ARIMA 模型被广泛运用在经济学、管理学、信息学及自然现象的预测上。

5.2.1 AR 模型

自回归模型(AutoRegressive model，简称 AR 模型)，是统计学中一种处理时间序列数据的方法，用同一变数(如 y_t 的前面各期，dm 即 y_1 至 y_{t-1})来预测本期(y_t)的表现，并假设它们为线性关系。因为这是从回归分析中的线性回归发展而来的，只是不用自变量预测 y_t，而是用 y_t 预测 y_{t+k}(自己)，所以称为自回归。那么序列元素 y_t 与其过去的依赖性就很重要，存在这种依赖性的简单例子是自回归过程。

$$y_t = \phi_1 y_{t-1} + u_t$$

便是这样一种一阶自回归模型，其中 u_t 为白噪声。

自回归模型描述{y_t}在某一时刻 t 和前 p 时刻序列值之间的线性关系，表示为

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + u_t$$

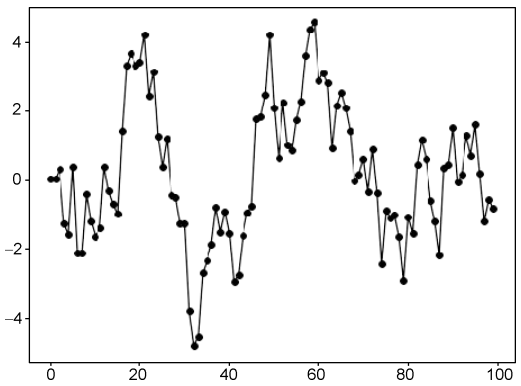
式中，随机序列{u_t}是白噪声，即 $u_t \sim N(0, \sigma^2)$ 且{u_t}与序列{y_t}($k < t$)不相关，该模型为 p 阶自回归模型，记为 AR(p)。实参数 $\phi_1, \phi_2, \cdots, \phi_p$ 称为自回归系数，是模型的待估参数。

模拟 AR(1)模型：$y_t = 0.8y_{t-1} + u_t$，$u_t \sim N(0,1)$。

In [9]

```
np.random.seed(123)    #种子数，确保每次模拟结果一样
n=100
y1=np.zeros(n);y1
u=np.random.randn(n);u
for t in range(2,n):
    y1[t]=0.8*y1[t-1]+u[t]
plt.plot(y1,'o-')
```

Out [9]



5.2.2 MR 模型

移动平均模型将序列{y_t}表示为白噪声的线性加权。

1 阶移动平均模型表示为

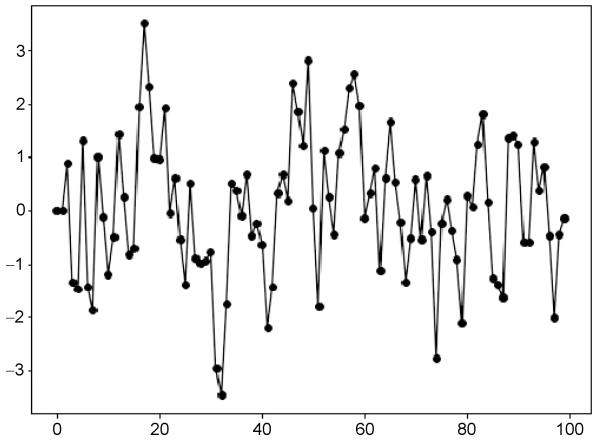
$$y_t = u_t + \theta_1 u_{t-1}$$

q 阶移动平均模型表示为

$$y_t = u_t+\theta_1 u_{t-1}+\theta_2 u_{t-2}+\cdots+\theta_q u_{t-q}$$

记为 MA(q)。实参数 θ_1，θ_1，…，θ_q 为移动平均系数，是模型的待估参数。

模拟 MR(1)模型：$y_t = u_t+0.6u_{t-1}$，$u_t \sim N(0,1)$。

In [10]	np.random.seed(123) y2=np.zeros(n);y2 u=np.random.randn(n);u for t in range(2,n): y2[t]=u[t]+0.6*u[t-1] plt.plot(y2,'o-')
Out [10]	

5.2.3 ARMA 模型

如果平稳随机过程既具有自回归过程的特性，又具有移动平均过程的特性，则不宜单独使用 AR(p)或 MA(q)模型，而需要两种模型混合使用。由于这种模型包含了自回归和移动平均两种成分，所以它的阶是二维的，由 p 和 q 两个数构成，其中 p 代表自回归成分的阶数，q 代表移动平均成分的阶数，记作 ARMA(p，q)，称作自回归移动平均混合模型或自回归移动平均模型。

自回归移动平均模型 ARMA(p,q)的一般表达式为

$$y_t = \phi_1 y_{t-1}+ \phi_2 y_{t-2}+\cdots+\phi_p y_{t-p} + u_t +\theta_1 u_{t-1}+\theta_2 u_{t-2}+\cdots+\theta_q u_{t-q}$$

显然，ARMA(0,q)=MA(q)，ARMA(p,0)= AR(p)，因此，MA(q)和 AR(p)可分别看作 ARMA(p,q)当 p=0 和 q=0 时的特例。

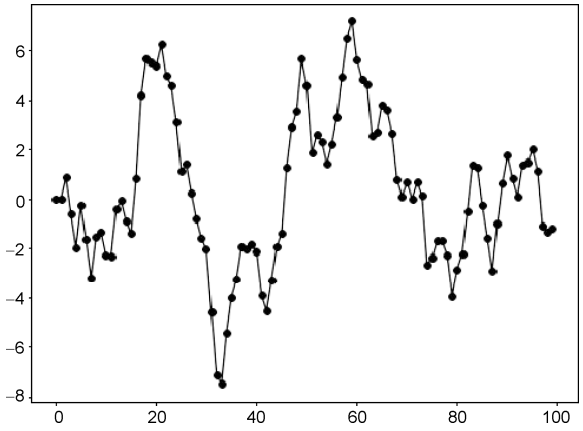
当 p=1，q=0 时为最简单的自回归模型，称为 AR(1)模型，即 $y_t = \phi_1 y_{t-1} + u_t$。其中 u_t 为误差项[这里通常假定为白噪声，即标准正态分布 $N(0,\sigma^2)$],ϕ_1 称为偏自相关系数。

当 p=0，q=1 时为最简单的移动自回归模型，称为 MA(1)模型：$y_t = u_t+\theta_1 u_{t-1}$。

当 p=1，q=1 时为最简单的自回归移动平均模型，称为 ARMA(1,1)模型：$y_t = \phi_1 y_{t-1} + u_t + \theta_1 u_{t-1}$。

ARMA(*p*,*q*)模型的优点是能以较少的参数描写单用 AR(*p*)或 MA(*q*)过程不能经济地描写的数据生成过程。在实际应用中，用 ARMA(*p*,*q*)拟合实际数据时所需阶数较低，*p* 和 *q* 的数值很少超过 2，因此，ARMA 模型在预测中具有很大的实用价值。

模拟 ARMA(1,1)模型：$y_t = 0.8y_{t-1} + u_t + 0.6u_{t-1}$。

In [11]	np.random.seed(123) y3=np.zeros(n);y3 u=np.random.randn(n);u for t in range(2,n): 　　y3[t]=0.8*y3[t−1]+u[t]+0.6*u[t−1] plt.plot(y3,'o-')
Out [11]	

5.2.4 ARIMA 模型

ARIMA(*p*,*d*,*q*)也称为差分自回归移动平均模型。AR 是自回归，*p* 为自回归项数；MA 为移动平均，*q* 为移动平均项数，*d* 为时间序列转化为平稳时间序列时所做的差分次数。所谓 ARIMA 模型，是指将非平稳时间序列转化为平稳时间序列，然后将因变量仅对它的滞后值及随机误差项的现值和滞后值进行回归所建立的模型。ARIMA 模型根据原序列是否平稳以及回归中所含部分的不同，包括移动平均过程(MA)、自回归过程(AR)、自回归移动平均过程(ARMA)及差分自回归移动平均过程(ARIMA)。

ARIMA 模型的基本思想是，将预测对象随时间推移而形成的数据序列视为一个随机序列，用一定的数学模型来近似描述该序列。该模型一旦被识别，就可以从时间序列的过去值及现在值来预测未来值。在某种程度上，现代统计方法、计量经济模型已经能够帮助企业对未来进行预测。

1 阶差分自回归移动平均模型：

$$\Delta y_t = \phi_1 \Delta y_{t-1} + \phi_2 \Delta y_{t-2} + \cdots + \phi_p \Delta y_{t-p} + u_t + \theta_1 u_{t-1} + \theta_2 u_{t-2} + \cdots + \theta_q u_{t-q}$$

d 阶差分自回归移动平均模型：

$$\Delta^d y_t = \phi_1 \Delta^d y_{t-1} + \phi_2 \Delta^d y_{t-2} + \cdots + \phi_p \Delta^d y_{t-p} + u_t + \theta_1 u_{t-1} + \theta_2 u_{t-2} + \cdots + \theta_q u_{t-q}$$

d 阶差分自回归移动平均模型记为 ARIMA(p,d,q)。d 为差分次数，$\phi_1,\phi_2,\cdots,\phi_d$ 为自回归系数，$\theta_1,\theta_2,\cdots,\theta_q$ 为移动平均系数，都是模型的待估参数。

显然，ARIMA(0,0,q)=MA(q)，ARMA(p,0,0)= AR(p)，ARMA(p,0,q)= ARMA(p,q)。

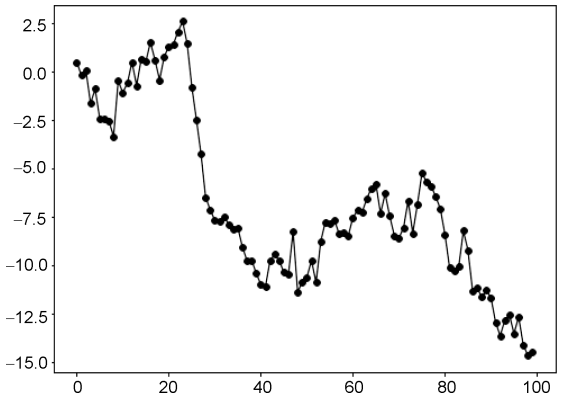
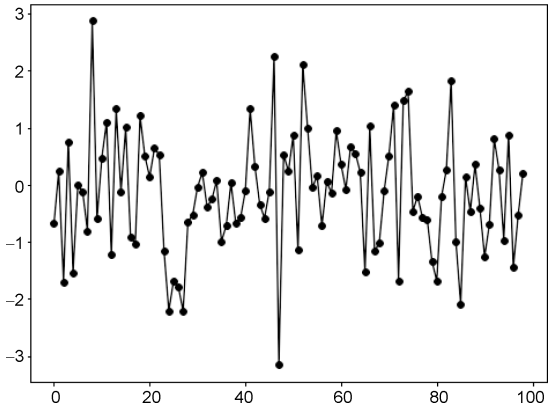
从前面的分析可知，实际中 ARIMA(p,d,q) 相当于 d 阶差分的 ARIMA(p,q)，所以，通常对差分数据建立 ARMA 模型即可获得 ARIMA 模型。d 阶差分定义为

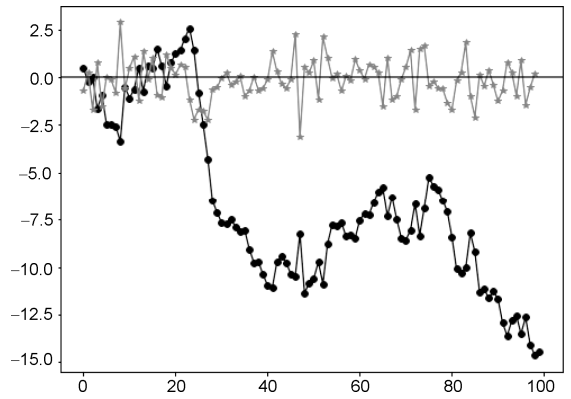
$$\Delta^d Y_t = Y_t - \Delta Y_{t-d}$$

式中，$\Delta^d Y_t$ 度量了 Y_t 与其滞后 Y_{t-d} 之间的差值。

Python 语言中定义了一个差分函数 diff，该函数的使用方法是 diff(x,d=1)，其中 d 表示差分阶数(默认为 1 阶)，即 diff(x) 表示 x 的一阶差分。

$$\Delta Y_t = Y_t - Y_{t-1}$$

In [12]	np.random.seed(12) n=100 y4=np.random.randn(n).cumsum() plt.plot(y4,'o-')
Out [12]	
In [13]	dy4=np.diff(y4) plt.plot(dy4,'o-')
Out [13]	
In [14]	plt.plot(y4,'o-',dy4,'*-');plt.axhline(0);

Out [14]	

对 Y_t 取一阶差分 $\Delta Y_t = Y_t - Y_{t-1} = u_t$，$\Delta Y_t$ 为白噪声，由于白噪声是一个平稳序列，所以序列 $\{\Delta Y_t\}$ 是平稳的。

5.3 ARMA 模型的构建

5.3.1 序列的相关性检验

5.3.1.1 自相关性检验

(1) 自相关系数计算

通常用自相关函数(auto correlation function，acf)来计算序列 y_t 中任意两个元素之间的自相关程度。对随机过程 $\{y_t\}$，样本 y_t 与 y_{t+k} 之间的自相关函数 $\text{acf}_k=\text{cor}(y_t,y_{t+k})$ 如下：

$$\hat{\rho}_k = \frac{\sum_{i=1}^{n-k}(y_t-\bar{y})(y_{t+k}-\bar{y})}{\sum_{i=1}^{n}(y_t-\bar{y})^2}$$

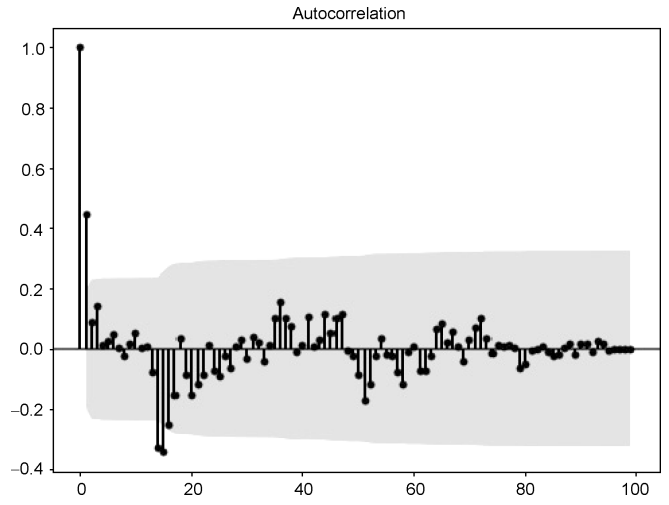
In [15]	from statsmodels.graphics.tsaplots import acf,plot_acf np.round(acf(y2),3)
Out [15]	array([1. , 0.447, 0.086, 0.142, 0.013, 0.028, 0.049, 0.001, −0.023, 0.018, 0.05, 0.001, 0.005, −0.077, −0.329, −0.341, −0.248, −0.154, 0.033, −0.086, −0.156, −0.121, −0.086, 0.011, −0.076, −0.093, −0.026, −0.065, 0.006, 0.028, −0.034, 0.04, 0.022, −0.042, 0.01, 0.102, 0.155, 0.101, 0.074, −0.01, 0.014])

(2) 自相关系数图示

随机时间序列模型着重研究的是相关关系，因此自相关函数在时间序列模型中占有重要地位，但要注意的是：在实际识别时，样本自相关函数 $\hat{\rho}_k$ 只是总体自相关函数 ρ_k 的估计，由于样本具有随机性，故当 $k > q$ 时，$\hat{\rho}_k$ 不会全为 0，而是在 0 的上下波动。

不过可以证明，当 $k > q$ 时，$\hat{\rho}_k$ 服从渐近正态分布，即 $\hat{\rho}_k \sim N(0,1/n)$，式中，$n$ 为样本容量。

因此，如果计算的 $\hat{\rho}_k$ 满足 $|\hat{\rho}_k| < 1.96/\sqrt{n}$，我们就有 95%的把握判断原时间序列不存在自相关性。

In [16]	plot_acf(y2); #MR(1)模型的自相关系数
Out [16]	

这里可以看出 MR 模型是 1 阶的，与我们的模拟一致。

(3) 自相关系数检验

① Box－Pierce 检验。

博克斯(G.E.P.Box)和皮尔斯(D.A.Pierce)提出的 Q 统计量可以检验时间序列的相关性。Q 统计量定义为

$$Q = n\sum_{k=1}^{m}\hat{\rho}_k^2$$

式中，n 为样本容量，m 为滞后长度。在大样本的情况下，它近似服从自由度为 m 的 χ^2 分布。若计算出的 Q 值大于在选定显著性水平下从 χ^2 分布表中查出的临界 Q 值，则拒绝所有真实的(r_k 都为 0)虚拟假设，这时序列不存在自相关性；否则，序列存在自相关性。

② Ljung–Box 检验。

1978 年扬–博克斯将博克斯和皮尔斯的 Q 统计量变形为 LB 统计量(Ljung-Box Statistic)，其定义为

$$\mathrm{LB} = n(n+2)\sum_{k=1}^{m}\left(\frac{\hat{\rho}_k^2}{n-k}\right) \sim \chi^2(m)$$

其检验过程与 Q 统计量的检验过程一样，但 LB 统计量比 Q 统计量有更好的小样本性质(即在统计意义上更有效)，所以 LB 统计量常用来检验小样本序列的相关性。如果其自相关系数都不显著，那么说明其序列不相关。

In [17]	def ac_QP(Yt):
	import statsmodels.api as sm
	r,q,p = sm.tsa.acf(Yt, qstat=True)
	rqp=np.c_[r[1:], q, p]
	rqp=pd.DataFrame(rqp, columns=["AC", "Q", "Prob(>Q)"]);
	return(rqp)
	
	ac_QP(y2)[:10]

Out [17]

	AC	Q	Prob(>Q)
0	0.4469	20.5762	5.7304e−06
1	0.0865	21.3546	2.3062e−05
2	0.1424	23.4861	3.1977e−05
3	0.0130	23.5039	1.0041e−04
4	0.0275	23.5853	2.6079e−04
5	0.0495	23.8510	5.5627e−04
6	0.0012	23.8511	1.2101e−03
7	−0.0229	23.9092	2.3735e−03
8	0.0180	23.9456	4.3881e−03
9	0.0503	24.2328	7.0059e−03

由于 Q 的 P 值都小于 0.05，于是拒绝原假设，认为序列存在一定的自相关性。

5.3.1.2 偏自相关性检验

(1) 偏自相关系数计算

自相关函数 acf_k 给出了 y_t 与 y_{t-k} 的总体相关性，但总体相关性可能掩盖了变量间不同的隐含关系。

与之相反，y_t 与 y_{t-k} 间的偏自相关函数(partial autocorrelation，简记为 pacf())则是消除了中间变量 y_{t-1}，y_{t-2}，…，y_{t-k+1} 带来的间接相关后的直接相关性，它是在已知序列值 y_{t-1}，y_{t-2}，…，y_{t-k+1} 的条件下，y_t 与 y_{t-k} 间关系的度量，即 $\text{pacf}_k = \text{cor}(y_t, y_{t-k})$，指在排除了 k 个中间变量 $y_{t-1}, y_{t-2}, \cdots, y_{t-k+1}$ 的影响后，y_t 和 y_{t-k} 的自相关系数。由于 acf() 和 pacf() 分别为 ϕ 和 θ 的函数，可据其初步判定时间序列所适合的阶数。

In [18]	from statsmodels.graphics.tsaplots import pacf,plot_pacf np.round(pacf(y1),3)pacf

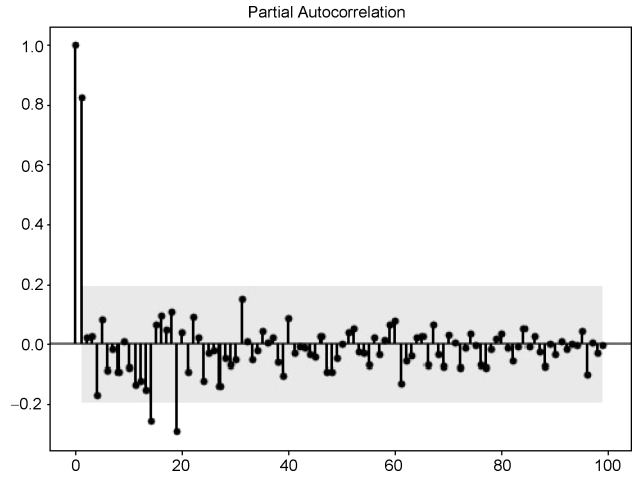
Out [18]

```
array([ 1.   ,  0.836,  0.025,  0.028, −0.183,  0.094, −0.097, −0.015,
       −0.109,  0.012, −0.094, −0.156, −0.152, −0.19 , −0.333,  0.056,
        0.115,  0.072,  0.158, −0.397,  0.04 , −0.161,  0.168,  0.043,
       −0.179, −0.056, −0.066, −0.242, −0.078, −0.173, −0.105,  0.311,
        0.006, −0.066, −0.122,  0.013,  0.04 ,  0.108, −0.134, −0.249,
        0.202])
```

(2) 偏自相关系数图示

要判断在 0.05 显著性水平下 pacf_k 是否为 0，只要考察其估计值是否落在区间

$[-1.96/\sqrt{n}, 1.96/\sqrt{n}]$ 内即可。如果估计值落在此区间内，则 $pacf_k$ 不显著，即确认 $pacf_k=0$；如果估计值落在此区间外，则 $pacf_k$ 显著，即确认 $pacf_k\neq0$。

In [19]	plot_pacf(y1)；　#AR(1)模型的偏自相关系数
Out [19]	

这里可以看出 AR 模型是 1 阶的，与我们的模拟一致。

5.3.2　ARMA 模型的建立与检验

运用 ARMA 模型的前提条件是：作为分析对象的时间序列是一组零均值的平稳随机序列。平稳随机序列的统计特性不随时间的推移而变化，直观地说，平稳随机序列的线图无明显的上升或下降趋势。下面对前面模拟的 ARMA(1,1)模型

$$y_t = 0.8y_{t-1} + u_t + 0.6u_{t-1}$$

进行建模。

5.3.2.1　模型阶数的识别

ARMA(p,q)模型应用的最大难点是阶数 p, q 的识别，一般根据时间序列的样本自相关函数(acf())、偏自相关函数(pacf())的特点来选择模型的类型。

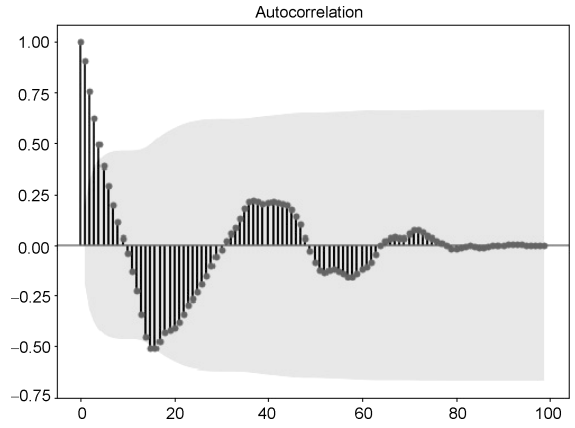
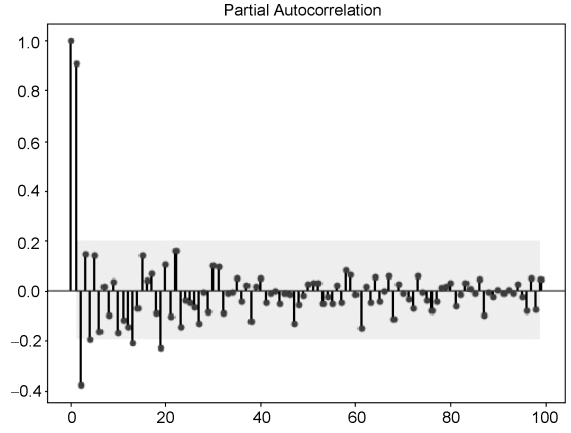
(1)相关图识别

如果序列$\{y_t\}$的自相关函数和偏自相关函数皆无截断点，即它们均是拖尾的，则可判定该序列为 ARMA 序列，如表 5-1 所示。

表 5-1　ARMA(p,q)模型的 acf()和 pacf()理论模式

模　　型	acf()	pacf()
AR(p)	衰减趋于 0 或振荡	p 阶截尾
MA(q)	q 阶截尾	衰减趋于 0 或振荡
ARMA(p,q)	q 阶后衰减趋于 0 或振荡	p 阶后衰减趋于 0 或振荡

具体见后面的自相关图和偏自相关图。

In [20]	plot_acf(y3);
Out [20]	
In [21]	plot_pacf(y3);
Out [21]	

用 acf() 和 pacf() 图示法确定 p 和 q 有时是不可靠的，因为很难同时满足要求。比如，我们从前面的自相关图和偏自相关图可以看到，p=2，q=4，即应该是 ARMA(2,4) 模型。

下面我们用信息量准则来确定 p 和 q 的阶数。

(2) 信息量识别

自相关函数和偏自相关函数只能初步断定序列$\{y_t\}$是否为 ARMA 模型(因为自相关函数和偏自相关函数都是拖尾的)，但不能确定其阶数，这时需要采用一定的定阶准则，目前选择模型常用如下准则：

* AIC = $-2\ln(L) + 2k$　　赤池信息量

* BIC = $-2\ln(L) + \ln(n) * k$　　贝叶斯信息量

* HQIC = $-2\ln(L) + \ln(\ln(n)) * k$　　汉南-奎因准则

这里 L 为模型的剩余残差平方和，n 为样本量，k 为与 p 和 q 相关的常数，对于由低阶到高阶不同的 p，q 取值，分别建立模型并进行参数估计，比较各模型的 AIC 值，使其达到最小的 p_0, q_0，这时的 (p_0,q_0) 为最佳模型阶数，构造这些统计量所遵循的统计思想是一致的，就是在考虑拟合残差的同时，依自变量个数施加“惩罚”。要注意的是，这些

准则不能说明某个模型的精确度，也就是说，对于三个模型 A，B，C，我们能够判断出 C 模型是最好的，但并不能保证 C 模型能够很好地刻画数据，因为有可能三个模型都是糟糕的。

ARMA 模型的阶数确定是一种困难的事，目前还没有最好的方法，statsmodels.tsa.stattools 包中的 arma_order_select_ic()函数提供了一种自动给出 ARMA(*p*,*q*)的 *p*，*q* 值的算法。下面分别对模拟的模型数据 y_1，y_2，y_3 分别确定其阶数。

In [22]	import statsmodels.tsa.stattools as ts ts.arma_order_select_ic(y1,max_ar=3,max_ma=3,ic=['aic','bic','hqic'])
Out [22]	'aic_min_order': (1, 0), 'bic_min_order': (1, 0) 'hqic_min_order': (1, 0)
In [23]	ts.arma_order_select_ic(y1,max_ar=3,max_ma=3,ic=['aic','bic','hqic'])
Out [23]	'aic_min_order': (0, 1), 'bic_min_order': (0, 1) 'hqic_min_order': (0, 1)
In [24]	ts.arma_order_select_ic(y3,max_ar=3,max_ma=3,ic=['aic', 'bic','hqic'])
Out [24]	'aic_min_order': (3, 2), 'bic_min_order': (1, 1) 'hqic_min_order': (3, 2)

根据 BIC 信息量准则，y_1 序列的阶数为(1,0)，为 AR(1)模型；y_2 序列的阶数为(0,1)，为 MR(1)模型；y_3 序列的阶数为(1,1)，为 ARMA(1,1)模型，自动给出了模型的参数 p=1，q=1，符合我们模拟的模型。

5.3.2.2 参数的估计与检验

ARMA 模型的参数估计方法较多，都可以按照线性回归模型思路去做，本书不进一步展开，计算过程颇为复杂，在实际工作中，一般使用通用软件进行估计。

(1)估计模型 AR(1)：$y_t = 0.8y_{t-1} + u_t$

In [25]	from statsmodels.tsa.arima_model import ARMA y1_arma=ARMA(y1,order=(1,0)).fit() y1_arma.summary()
Out [25]	ARMA Model Results ========== Dep. Variable: y No. Observations: 100 Model: ARMA(1, 0) Log Likelihood −153.530 Method: css-mle S.D. of innovations 1.117 Date: Sat, 02 Jun 2018 AIC 313.060 Time: 15:35:02 BIC 320.875 Sample: 0 HQIC 316.223 ========== coef std err z P>|z| [0.025 0.975]

	——						
	const	0.1233	0.601	0.205	0.838	−1.054	1.300
	ar.L1.y	0.8220	0.055	15.027	0.000	0.715	0.929

估计的结果和模拟模型 AR(1)：$y_t = 0.8y_{t-1} + u_t$ 基本吻合，常数项不显著。

(2) 估计模型 MA(1)：$y_t = u_t + 0.6u_{t-1}$

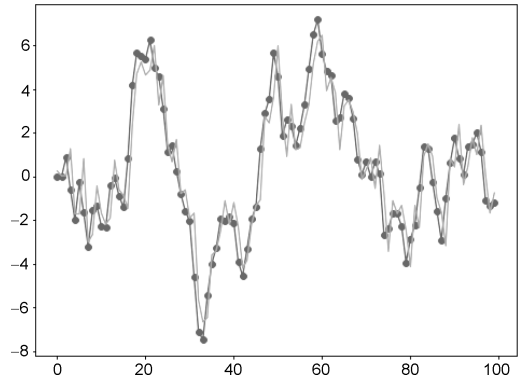
In [26]	ARMA(y2,order=(0,1)).fit().summary()					
Out [26]	ARMA Model Results					
	==========					
	Dep. Variable:		y	No. Observations:		100
	Model:		ARMA(0, 1)	Log Likelihood		−153.165
	Method:		css-mle	S.D. of innovations		1.115
	Date:		Sat, 02 Jun 2018	AIC		312.331
	Time:		15:46:55	BIC		320.146
	Sample:		0	HQIC		315.494
	==========					
		coef	std err	z	P>|z|	[0.025	0.975]
	——						
	const	0.0521	0.189	0.275	0.784	−0.319	0.423
	ma.L1.y	0.7051	0.089	7.966	0.000	0.532	0.879

估计的结果和模拟模型 MA(1)：$y_t = u_t + 0.6u_{t-1}$ 基本吻合，常数项不显著。

(3) 估计 ARMR(1,1) 模型：$y_t = 0.8y_{t-1} + ut + 0.6u_{t-1}$

In [27]	ARMA(y3,order=(1,1)).fit().summary()					
Out [27]	ARMA Model Results					
	==========					
	Dep. Variable:		y	No. Observations:		100
	Model:		ARMA(1, 1)	Log Likelihood		−154.080
	Method:		css-mle	S.D. of innovations		1.115
	Date:		Sat, 02 Jun 2018	AIC		316.160
	Time:		15:50:18	BIC		326.580
	Sample:		0	HQIC		320.377
	==========					
		coef	std err	z	P>|z|	[0.025	0.975]
	——						
	const	0.2466	0.901	0.274	0.785	−1.518	2.012
	ar.L1.y	0.7977	0.062	12.924	0.000	0.677	0.919
	ma.L1.y	0.7029	0.095	7.395	0.000	0.517	0.889

估计的结果和模拟 ARMR(1,1) 模型 $y_t = 0.8y_{t-1} + u_t + 0.6u_{t-1}$ 基本吻合，常数项不显著。

In [28]	plt.plot(y3,'o-',ARMA(y3,order=(1,1)).fit().fittedvalues);
Out [28]	

ARMA 模型的参数检验和模型检验类似于回归模型的参数检验和模型检验，原理和公式相对比较复杂，在此予以省略，可参考相关文献。

5.3.3 序列的平稳性检验

在实际中遇到的时间序列数据很可能是非平稳序列，而平稳性在计量经济建模中又具有重要地位，因此有必要对观测值的时间序列数据进行平稳性检验。

对时间序列的平稳性除了通过散点图直观判断外，运用统计量进行统计检验则是更为准确与重要的。单位根检验是平稳性检验中普遍应用的一种统计检验方法。单位根检验是建立 ARMA 模型、ARIMA 模型，变量间的协整分析，因果关系检验等的基础。

5.3.3.1 单位根检验(DF)

我们已知道，随机游走序列 $Y_t = Y_{t-1} + u_t$ 是非平稳的，其中 u_t 是白噪声。而该序列可看成 1 阶自回归 AR(1)过程 $Y_t = \rho Y_{t-1} + u_t$ 中参数 ρ=1 时的特例。也就是说，对式

$$Y_t = \rho Y_{t-1} + u_t$$

做回归，如果确实发现 ρ=1，就说随机变量 Y_t 有一个单位根。可变形成差分形式：

$$\Delta Y_t = (1-\rho) Y_{t-1} + u_t = \delta Y_{t-1} + u_t$$

检验是否存在单位根 ρ=1，也可通过判断是否有 δ=0 来实现。

一般地，检验一个时间序列 Y_t 的平稳性，可通过检验带有截距项的一阶自回归模型

$$Y_t = \alpha + \rho Y_{t-1} + u_t$$

中的参数 ρ 是否小于 1 来实现，或者检验其等价变形式

$$\Delta X_t = \alpha + \delta X_{t-1} + u_t$$

中的参数 δ 是否小于 0。

可以证明，当参数 $\rho > 1$ 或 $\rho = 1$ 时，时间序列是非平稳的，即 δ>0 或 δ=0。因此，我们关心的检验为

原假设 H_0：$\delta = 0$；备择假设 H_1：$\delta < 0$。

上述检验可通过 OLS 法下的 t 检验完成。

然而，在原假设(序列非平稳)下，即使在大样本下，t 统计量也是有偏误的(向下偏倚)，通常的 t 检验无法使用。

5.3.3.2 扩展单位根检验(ADF)

DF 检验存在的问题是，在检验所设定的模型时，假设随机扰动项不存在自相关性，但大多数经济数据序列是不能满足此项假设的，当随机扰动项存在自相关时，直接使用 DF 检验法会出现偏差，为了保证单位根检验的有效性，人们对 DF 检验进行拓展，从而形成了扩展的 DF 检验(Augmented Dickey-Fuller test)，简称为 ADF 检验。

在上述使用 $\Delta Y_t = \alpha + \delta Y_{t-1} + u_t$ 对时间序列进行的平稳性检验中，实际上假定了时间序列是由具有白噪声随机误差项的一阶自回归过程 AR(1)生成的，但在实际检验中，时间序列可能由更高阶的自回归过程生成，或者随机误差项并非白噪声，这样用 OLS 法进行估计均会表现出随机误差项自相关，导致 DF 检验无效。

另外，如果时间序列包含明显的随时间变化的某种趋势(如上升或下降)，则也容易导致上述检验中的自相关随机误差项问题。

为了保证 DF 检验中随机误差项的白噪声特性，Dicky 和 Fuller 对 DF 检验进行了扩充，形成了 ADF 检验。

ADF 检验的基本模型为

$$\text{模型 1：}\quad Y_t = \gamma Y_{t-1} + \varepsilon_t$$

$$\text{模型 2：}\quad Y_t = \alpha + \gamma Y_{t-1} + \varepsilon_t$$

$$\text{模型 3：}\quad Y_t = \alpha + \beta t + \gamma Y_{t-1} + \varepsilon_t$$

式中，ε_t 为随机扰动项。

模型 3 中的 t 是时间变量，代表了时间序列随时间变化的某种趋势(如果有的话)。检验的假设都是，针对 H_1: $\gamma < 0$，检验 H_0: $\gamma = 0$，即存在一个单位根。模型 1 与另两个模型的差别在于，是否包含常数项和趋势项。实际检验时从模型 3 开始，然后模型 2、模型 1。何时检验拒绝原假设，即原序列不存在单位根，为平稳序列，何时检验停止；否则，继续检验，直到检验完模型 1 为止。

一个简单的检验过程：同时估计出上述三个模型的适当形式，然后通过 ADF 临界值表检验原假设 H_0：$\gamma = 0$。

① 只要其中有一个模型的检验结果拒绝了原假设，就可以认为时间序列是平稳的；

② 当三个模型的检验结果都不能拒绝原假设时，则认为时间序列是非平稳的。

这里所谓模型适当的形式指，在每个模型中选取适当的滞后差分项，以使模型的残差项是一个白噪声(主要保证不存在自相关)。

可以验证，y_1，y_2，y_3 都是平稳时间序列，下面使用 ADF 方法验证 y_4 和 dy_4 的平稳性。

In [29]	from statsmodels.tsa.stattools import adfuller def ADF(ts): #平稳性检验

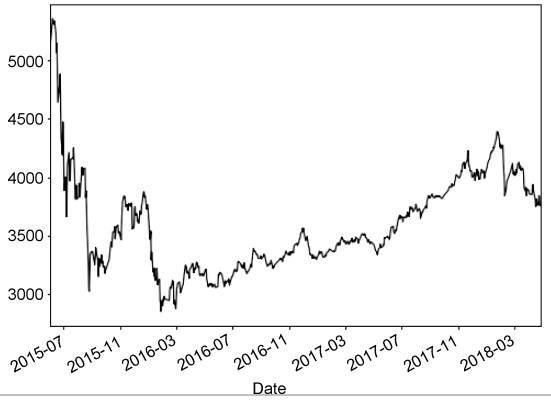
	dftest = adfuller(ts) dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value', '#Lags Used','Number of Observations Used']) for key,value in dftest[4].items(): dfoutput['Critical Value (%s)'%key] = value return round(dfoutput, 4) ADF(y4)
Out [29]	Test Statistic –1.0933 p-value 0.7177 #Lags Used 0.0000 Number of Observations Used 99.0000 Critical Value (1%) –3.4982 Critical Value (5%) –2.8912 Critical Value (10%) –2.5826
In [30]	ADF(dy4)
Out [30]	Test Statistic –10.4611 p-value 0.0000 #Lags Used 0.0000 Number of Observations Used 98.0000 Critical Value (1%) –3.4989 Critical Value (5%) –2.8915 Critical Value (10%) –2.5828

从检验结果可以看到，序列 y_4 是非平稳序列，而差分序列 dy_4 是平稳序列，符合我们模拟的设定。

对平稳时间序列模型，可按前面的方法建立相应的 ARMA 模型进行分析。

5.4 股票指数预测模型的构建

下面针对沪深股票指数收盘价建立 ARIMA 模型，取 2015 年 4 月至 2018 年 4 月的数据作为训练样本，2018 年 5 月的数据作为预测对照样本。

In [31]	Ct=TSdata['2015-04':'2018-04'].Close; Ct.plot()
Out [31]	

5.4.1 模型的预处理

(1)预处理：平稳性验证

运用博克斯-詹金斯法的前提条件是，作为分析对象的时间序列是一组零均值的平稳随机序列。平稳随机序列的统计特性不随时间的推移而变化。直观地说，平稳随机序列的线图无明显的上升或下降趋势；但是，大量的社会经济现象随时间的推移，总表现出某种上升或下降的趋势，构成非零均值的非平稳的时间序列。从上页图中可看出，2016年的调整价基本是一个平稳序列，下面用 ADF 方法进行检验。

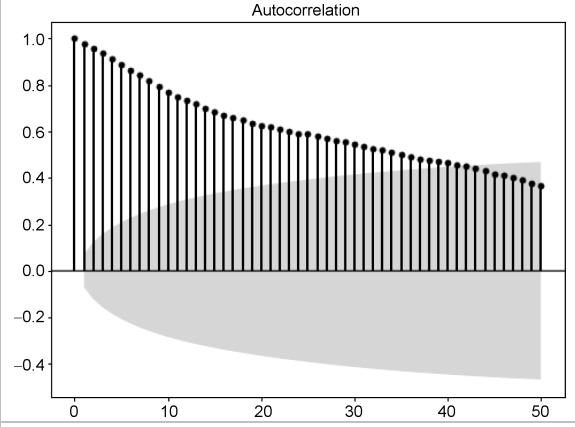
In [32]	ADF(Ct)	
Out [32]	Test Statistic	−3.7320
	p-value	0.0037
	#Lags Used	14.0000
	Number of Observations Used	694.0000
	Critical Value (1%)	−3.4398
	Critical Value (5%)	−2.8657
	Critical Value (10%)	−2.5690

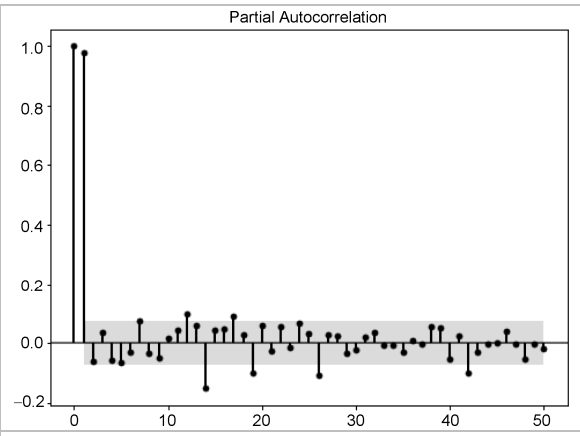
检验结果页证明了 Ct 序列是一个平稳时间序列。

(2)模型阶选择

首先，作出相应的时间序列图，判断时间序列有无上升或下降趋势、异常点、缺失点和结构变化。如存在趋势，则需进行差分；异常点则需要修正或者去除等。然后作样本自相关函数图和偏自相关函数图，并与理论 ARMA 模型的自相关函数和偏自相关函数图进行比较，选择可能合适的模型。模型选择的准则可以采用 AIC 或 BIC 等准则。

博克斯-詹金斯法是以时间序列的自相关分析为基础的，以便识别时间序列的模式，实现建模的任务。

In [33]	plot_acf(Ct,lags=50);
Out [33]	
In [34]	plot_pacf(Ct,lags=50);

Out [34]	
In [35]	import statsmodels.tsa.stattools as ts ts.arma_order_select_ic(Ct,max_ar=3,max_ma=3,ic=['aic','bic','hqic'])
Out [35]	{'aic': 0 1 2 3 0 10563.7376 9692.3102 10208.9686 9221.9458 1 7829.7406 7825.9016 7820.6973 7822.6958 2 7827.2676 7821.4259 7822.6970 7822.0358 3 7819.0652 7820.9179 NaN NaN, 'aic_min_order': (3, 0), 'bic': 0 1 2 3 0 10572.8653 9706.0017 10227.2241 9244.7650 1 7843.4321 7844.1570 7843.5166 7850.0790 2 7845.5231 7844.2452 7850.0801 7853.9828 3 7841.8844 7848.3010 NaN NaN, 'bic_min_order': (3, 0), 'hqic': 0 1 2 3 0 10567.2639 9697.5996 10216.0213 9230.7615 1 7835.0300 7832.9542 7829.5131 7833.2748 2 7834.3203 7830.2417 7833.2759 7834.3779 3 7827.8809 7831.4968 NaN NaN, 'hqic_min_order': (3, 0)}

根据信息量准则，可选取 ARMA(3,0)模型。

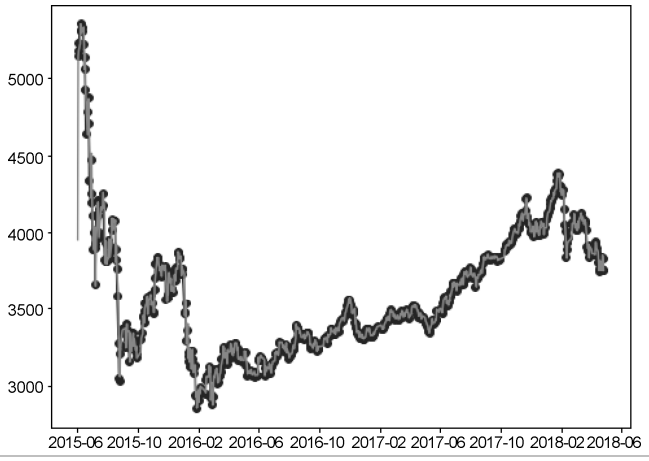
5.4.2 参数的估计与检验

ARMA 模型的参数估计方法较多，但都可以按照线性回归模型思路去做，本书不进一步展开。而 ARMA(p,q)的参数估计，需要用非线性估计法，计算过程颇为复杂，在实际工作中，一般使用通用软件进行估计。

In [36]	from statsmodels.tsa.arima_model import ARMA Ct_ARMA=ARMA(Ct,order=(3,0)).fit() Ct_ARMA.summary()
Out [36]	ARMA Model Results ==========

	Dep. Variable:		Close	No. Observations:		709
	Model:		ARMA(3, 0)	Log Likelihood		−3904.533
	Method:		css-mle	S.D. of innovations		59.429
	Date:		Sat, 02 Jun 2018	AIC		7819.065
	Time:		18:32:47	BIC		7841.884
	Sample:		06-03-2015 - 04-27-2018	HQIC		7827.881
		coef	std err	z	P>|z|	[0.025 0.975]
	const	3950.7953	453.051	8.720	0.000	3062.832 4838.759
	ar.L1.Close	1.0834	0.037	29.079	0.000	1.010 1.156
	ar.L2.Close	−0.2076	0.055	−3.799	0.000	−0.315 −0.101
	ar.L3.Close	0.1201	0.037	3.208	0.001	0.047 0.193

ARMA 模型的参数检验和模型检验类似于回归模型的参数检验和模型检验，原理和公式相对较复杂，在此予以省略，可参考相关文献。下面是拟合效果图，从图中可以看出，不考虑其他因素影响，我们建立的单纯的沪深 300 收盘价指数模型还是很不错的。

In [37]	plt.plot(Ct,'o-',Ct_ARMA.fittedvalues);
Out [37]	

5.4.3 模型的预测

建立模型的一大用途就是用其进行预测。若实际的数据生成过程是已知的，并且$\{y_t\}$和$\{u_t\}$的现在和过去各期的数值也已知，则可以以现在为原点，根据已掌握的信息，使用条件期望的方法对序列$\{y_t\}$未来各期数值进行预测。与回归分析模型一样，Python 中进行模型预测的函数为 forecast，但这里是没有自变量的，是向前预测期数。

In [38]	Ct_05=pd.DataFrame({'实际值':TSdata['2018-05'].Close});	#2018-05 收盘价数据
	Ct_05['预测值']=Ct_ARMA.forecast(22)[0]	#模型预测数据
	Ct_05['绝对误差']=Ct_05['实际值']-Ct_05['预测值'];	
	Ct_05['相对误差(%)']=Ct_05['绝对误差']/Ct_05['实际值']*100;	
	Ct_05	

Out [38]

Date	实际值	预测值	绝对误差	相对误差(%)
2018-05-02	3763.65	3766.6043	−2.9543	−0.0785
2018-05-03	3793.00	3768.0568	24.9432	0.6576
2018-05-04	3774.60	3767.7781	6.8219	0.1807
2018-05-07	3834.19	3768.3427	65.8473	1.7174
2018-05-08	3878.68	3769.1867	109.4933	2.8230
2018-05-09	3871.62	3769.9503	101.6697	2.6260
2018-05-10	3893.06	3770.6702	122.3898	3.1438
2018-05-11	3872.84	3771.3929	101.4471	2.6195
2018-05-14	3909.29	3772.1180	137.1720	3.5089
2018-05-15	3924.10	3772.8400	151.2600	3.8546
2018-05-16	3892.84	3773.5585	119.2815	3.0641
2018-05-17	3864.05	3774.2740	89.7760	2.3234
2018-05-18	3903.06	3774.9866	128.0734	3.2814
2018-05-21	3921.24	3775.6965	145.5435	3.7117
2018-05-22	3906.21	3776.4034	129.8066	3.3231
2018-05-23	3854.58	3777.1075	77.4725	2.0099
2018-05-24	3827.22	3777.8087	49.4113	1.2910
2018-05-25	3816.50	3778.5071	37.9929	0.9955
2018-05-28	3833.26	3779.2027	54.0573	1.4102
2018-05-29	3804.01	3779.8955	24.1145	0.6339
2018-05-30	3723.37	3780.5855	−57.2155	−1.5367
2018-05-31	3802.38	3781.2727	21.1073	0.5551

可以看出，模型的预测效果还是不错的。

数据与练习 5

5.1 AirPassengers 数据集(datasets 包)包含了 1949—1960 年间月度国际航班乘客总人数的数据。该数据是时间序列格式，单位为千人。

(1)请画该数据的线图。

(2)试分别构建 AR、MR、ARMA 和 ARIMA 模型。

5.2 EuStockMarkets 数据集(datasets 包)包含了 1991—1998 年间欧洲主要股票交易市场的日收盘价。

该数据是时间序列格式，由 1860 行、4 个变量构成。4 个变量分别代表欧洲的 4 个主要股票市场：Germany DAX (Ibis)，Switzerland SMI，France CAC，UK FTSE。

(1)请画该数据的线图。

(2)试分别构建 AR、MR、ARMA 和 ARIMA 模型。

5.3 Johnson 数据集(datasets 包)包含强生公司 1960—1980 年间的季度收入。该数

据是时间序列格式。

(1)请画该数据的线图。

(2)试分别构建 AR、MR、ARMA 和 ARIMA 模型。

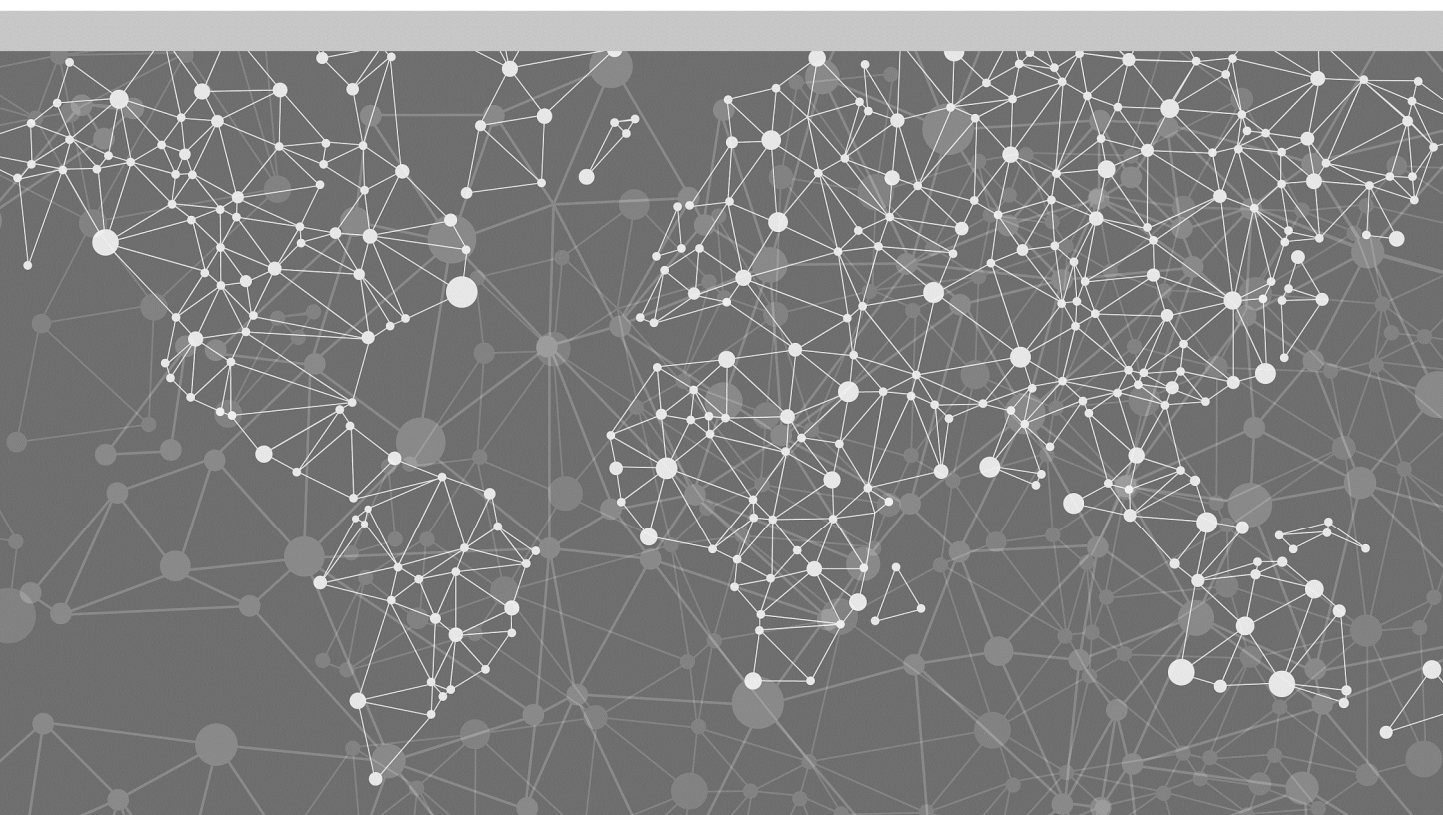
5.4 对全国居民消费价格指数进行分析。请读者从 Tushare 网站(http://tushare.org/)选取 2000 年 1 月至 2018 年 12 月的全国居民消费价格指数 CPI(月度数据，上年同月=100)作为样本数据，用 Python 语言命令进行数据分析，并建立相应的预测模型。

5.5 股票收益率的研究。请读者从 Tushare 网站(http://tushare.org/)选取 2015 年 1 月 1 日至 2018 年 12 月 31 日的沪深 300 指数作为样本数据，对我国证券市场沪深 300 股票指数收益率的变动进行分析，并用 Python 语言命令建立相应的模型，从中选取一个合适的模型。

第三部分

大数据

基本处理方法



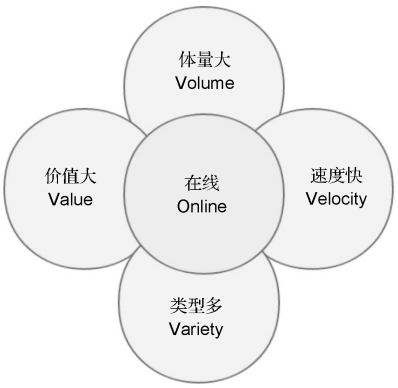
第 6 章　大数据分析基础应用

6.1　大数据的概念

6.1.1　大数据的含义

最早提出大数据时代到来的是麦肯锡：“数据，已经渗透到当今每个行业和业务职能领域，成为重要的生产因素。人们对于海量数据的挖掘和运用，预示着新一波生产率增长和消费者盈余浪潮的到来。”

业界(IBM 最早定义)将大数据的特征归纳为 4 个“V”，即体量大(Volume)、速度快(Velocity)、类型多(Variety)、价值大(Value)，或者说特点有四个层面：第一，数据体量巨大，大数据的起始计量单位至少是 PB(10^3TB)、EB(10^5TB)或 ZB(10^9TB)；第二，数据类型繁多，比如网络日志、视频、图片、地理位置信息等；第三，价值密度低，商业价值高，须进行数据挖掘。第四，数据收集频率高，维度大，处理速度快。最后一点和传统的数据分析技术有着本质的不同。



大数据的主要内容

大数据正在不断改变着人们的生活，在未来一段时间内，大数据将成为企业、社会和国家层面重要的战略资源。大数据将不断成为各类机构(尤其是企业)的重要资产，成为提升机构和公司竞争力的有力武器。企业将更加“钟情”于用户数据，充分利用客户与其在线产品或服务交互产生的数据，并从中获取价值。此外，在市场影响方面，大数据也将扮演重要角色——影响着广告、产品推销和消费者行为。

数据科学作为一个与大数据相关的新兴学科的出现，促进了大量的数据科学类专著

的出版。大数据也将催生一批新的就业岗位，如数据分析师、数据科学家等。具有丰富经验的数据分析人才成为稀缺资源，数据驱动型工作机会将呈现出爆炸式的增长。

6.1.2 大数据应用举例

近两年，“大数据”这个词越来越为大众所熟悉，“大数据”一直以“高冷”的形象出现在大众面前，面对大数据，许多人都一头雾水。下面通过几个经典案例，让大家实打实地“触摸”一把“大数据”。你会发现它其实就在身边，而且也是很有趣的。

(1)啤酒与尿布

全球零售业巨头沃尔玛在对消费者购物行为分析时发现，男性顾客在购买婴儿尿片时，常常会顺便搭配几瓶啤酒来犒劳自己，于是尝试推出了将啤酒和尿布摆在一起的促销手段。没想到这个举措居然使尿布和啤酒的销量都大幅增加了。如今，“啤酒+尿布”的数据分析成果已成为大数据技术应用的经典案例，被人津津乐道。

(2)数据新闻让英国撤军

2010 年 10 月 23 日，《卫报》利用维基解密的数据做了一篇“数据新闻”，将伊拉克战争中所有的人员伤亡情况标注于地图之上，地图上一个红点便代表一次死伤事件，鼠标点击红点后弹出的窗口有详细的说明：伤亡人数、时间、造成伤亡的具体原因。密布的红点多达 39 万个，触目惊心。该文一经刊出立即引起震动，最终推动英国做出撤出驻伊拉克军队的决定。

(3)“魔镜”预知石油市场走向

如果你对“魔镜”的认知还停留在“魔镜魔镜，告诉我谁是世界上最美的女人”，那你就真的落伍了。“魔镜”不仅是童话中王后的宝贝，而且是现实世界中的一款神器。其实，“魔镜”是苏州国云数据科技有限公司的一款出色的大数据可视化产品，而且是国内首款。现在，“魔镜”通过数据的整合分析可视化不仅可以得出“谁是世界上最美的女人”，还能通过价量关系得出市场的走向。“魔镜”曾帮助中石化等企业分析数据，将数据可视化，使企业科学地进行判断、决策，节约了成本，合理配置了资源，提高了收益。

未来大数据的应用场景主要集中于以下几方面：

① 利用大数据实现客户交互改进。电信、零售、旅游、金融服务和汽车等行业将“快速抓取客户信息从而了解客户需求”列为首要任务。

② 利用大数据实现运营分析优化。制造、能源、公共事业、电信、旅行和运输等行业要时刻关注突发事件，通过监控提升运营效率并预测潜在风险。

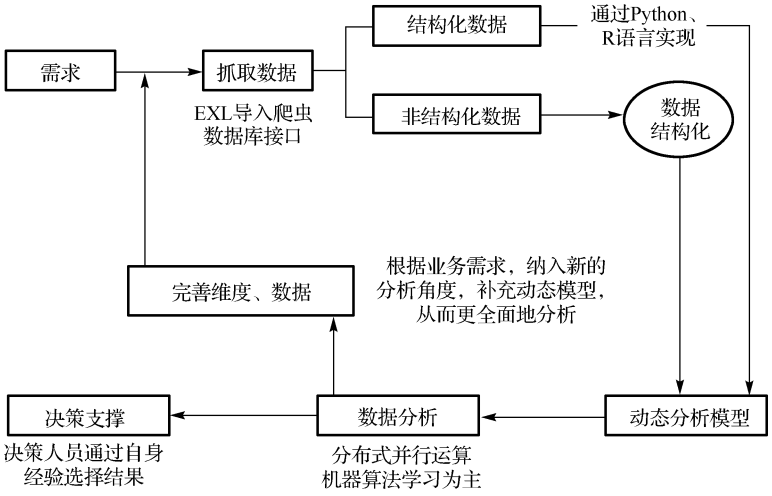
③ 利用大数据实现 IT 效率和规模效益。企业需要增强现有数据仓库基础架构，实现大数据传输、低延迟和查询的需求，确保有效利用预测分析和商业智能实现或扩展某些性能。

④ 利用大数据实现智能安全防范。政府、保险等行业亟待利用大数据技术补充和加强传统的安全解决方案。

当然，不论哪个行业的大数据分析和应用场景，其典型特点之一都是，无法离开以人为中心所产生的各种用户行为数据、用户业务活动和交易记录、用户社交数据，这些核心数据的相关性再加上可感知设备的智能数据采集，就构成一个完整的大数据生态环境。

6.1.3 大数据分析方法

越来越多的应用涉及大数据，这些大数据的属性包括数量、速度、多样性等，都呈现了大数据不断增长的复杂性，所以，大数据的分析在大数据领域就显得尤为重要，可以说是决定最终信息是否有价值的决定性因素。基于此，大数据分析的方法理论有哪些呢？



大数据时代的数据分析体系

6.1.3.1 基础知识

大数据分析方法的基础知识主要有：

① 数据库基本知识；

② 数学及编程能力；

③ 统计理论与相关知识。

6.1.3.2 基本方面

大数据分析的基本方面主要有：

(1)预测性分析能力

数据挖掘可以让分析员更好地理解数据，而预测性分析可以让分析员根据可视化分析和数据挖掘的结果做出一些预测性的判断。

(2)数据质量和数据管理

数据质量和数据管理是一些管理方面的最佳实践。通过标准化的流程和工具对数据进行处理，可以保证一个预先定义好的高质量的分析结果。

(3)可视化分析

不管是对数据分析专家还是普通用户而言，数据可视化都是数据分析工具最基本的

要求。可视化可以直观展示数据，让数据“自己说话”，让观众“听”到结果。

(4)语义引擎

非结构化数据的多样性带来了数据分析的新挑战，需要一系列工具去解析、提取、分析数据，语义引擎要设计成能够从“文档”中智能提取信息。

(5)数据挖掘算法

可视化是给人看的，数据挖掘就是给机器“看”的。集群、分割、孤立点分析以及其他算法让我们深入数据内部挖掘价值。这些算法不仅要处理大数据的量，也要处理大数据的速度。

6.1.3.3 大数据处理

大数据处理时代理念的三大转变：要全体而不只是样本，要效率而不仅是绝对精确，要相关性也要因果关系。具体的大数据处理方法有很多，根据长时间的实践，整个处理流程可以概括为四步，分别是采集、导入和预处理、统计和分析，以及挖掘。

(1)采集(存储)

大数据的采集指利用多个数据库来接收发自客户端的数据，并且用户可以通过这些数据库进行简单的查询和处理工作。比如，电商使用传统的关系型数据库 MySQL 和 Oracle 等来存储每一笔事务数据，除此之外，Redis 和 MongoDB 这样的非结构化数据库也常用于数据的采集。

在大数据的采集过程中，其主要特点和挑战是并发数高，因为可能同时会有成千上万个用户进行访问和操作，比如火车票售票网站和淘宝网，它们并发的访问量在峰值时达到上百万，所以需要在采集端部署大量数据库才能支撑。如何在这些数据库之间进行负载均衡和分片？这需要深入思考和设计。

(2)导入和预处理

虽然采集端本身会有很多数据库，但如果要对这些海量数据进行有效的分析，还是应该将这些来自前端的数据导入一个集中的大型分布式数据库或者分布式存储集群，并且可以在导入基础上做一些简单的清洗和预处理工作。也有一些用户会在导入时使用来自 Twitter 的 Storm 对数据进行流式计算，来满足部分业务的实时计算需求。导入与预处理过程的特点和挑战主要是导入的数据量大，每秒钟的导入量经常会达到百兆，甚至千兆级别。

(3)统计和分析

统计和分析主要利用分布式数据库，或者分布式计算集群对存储于其内的海量数据进行普通的分析和分类汇总等，以满足大多数常见的分析需求，在这方面，一些实时性需求会用到 EMC 的 GreenPlum、Oracle 的 Exadata，以及基于 MySQL 的列式存储 Infobright 等，而一些批处理或者基于半结构化数据的需求可以使用 Hadoop。统计与分析部分的主要特点和挑战是涉及的数据量大，对系统资源，特别是 I/O，占用量极大。

(4)挖掘(算法)

与前面的统计和分析过程不同的是，数据挖掘一般没有什么预先设定好的主题，主要是对现有数据进行基于各种算法的计算，从而达到预测的效果，并实现一些高级别数

据分析的需求。比较典型的算法有用于聚类的 K-Means 算法、用于统计学习的 SVM 算法和用于分类的 Naive Bayes 算法等，主要使用的工具有 Hadoop、Mahout 等。该过程的特点和挑战主要是用于挖掘的算法很复杂，并且计算涉及的数据量和计算量都很大，此外，常用的数据挖掘算法都以单线程为主。

6.2 Python 文本预处理

Python 中文文献计量分析没有现成的函数与方法，可以根据 Python 自带的字符处理函数编写文献计量分析所需要的函数。首先将文献题录数据当成一般的中文文本数据集，根据其自身特征进行文本预处理。这里介绍一些常用且简单的 Python 字符处理函数，掌握它们之后做文献计量分析就得心应手了。

6.2.1 字符串的基本操作

6.2.1.1 字符及字符串统计

直接使用 len()函数可分别对字段自身长度、列表长度和嵌套列表长度进行统计，len()函数也可以直接对中文字段进行操作。

In [1]	len('abc')
Out [1]	3
In [2]	S=["asfef", "qwerty", "yuiop", "b", "stuff.blah.yech"]; len(S)
Out [2]	5
In [3]	[len(s) for s in S]
Out [3]	[5, 6, 5, 1, 15]

6.2.1.2 字符串连接与拆分

(1)连接方法 1：加号'+'

直接使用加号'+'就可以实现对两个或多个字符串进行连接。

In [4]	'Python'+' '+'Data Analysis' '暨南大学'+'管理学院'
Out [4]	'Python Data Analysis' '暨南大学管理学院'

(2)连接方法 2：字符串格式化输出

有时对连接有自定义操作，这时可以采用字符串格式化输出，这种方法更为常用。

In [5]	website = '%s%s%s' % ('Python', 'tab', '.com');website
Out [5]	'Pythontab.com'

(3)连接方法 3：join()

如果操作的对象是列表，也可以采用 join()函数。

In [6]	listStr = ['Python', 'tab', '.com'] ''.join(listStr) #paste
Out [6]	'Pythontab.com'

(4)拆分方法：split()

选择 2017 年 8 月出版的《中国社会科学》前三篇文章的题录数据进行中文文本处理，如表 6-1 所示。通过学习这部分内容，可以为下一部分做文献计量分析打好重要的编程基础，全部函数操作都来源于这一节。Python 内置针对字段的拆分函数 split()。

表 6-1 《中国社会科学》前三篇文章的题录数据

Title-题名	Author-作者	Organ-单位	Source-文献来源	Keyword-关键词
历史阐释中的历史事实和历史评价问题——基于马克思唯物历史观的基本理论和方法	涂成林	广州大学广州发展研究院	中国社会科学	历史阐释；历史事实；历史评价；唯物史观
钦差巡察与查理曼的帝国治理	李云飞	暨南大学文学院历史系	中国社会科学	查理曼；钦差巡察；加洛林帝国；法兰克；中世纪
南宋史料与政治史研究——三重视角的分析	黄宽重	台湾长庚大学/长庚医院	中国社会科学	南宋；政治忌讳；人物评价；人际关系；包容政治

In [7]	S1='历史阐释;;历史事实;;历史评价;;唯物史观' S1.split(';;')
Out [7]	['历史阐释', '历史事实', '历史评价', '唯物史观']
In [8]	S2='查理曼;;钦差巡察;;加洛林帝国;;法兰克;;中世纪' S3='南宋;;政治忌讳;;人物评价;;人际关系;;包容政治' S4=[S1,S2,S3];S4
Out [8]	['历史阐释;;历史事实;;历史评价;;唯物史观', '查理曼;;钦差巡察;;加洛林帝国;;法兰克;;中世纪', '南宋;;政治忌讳;;人物评价;;人际关系;;包容政治']

针对列表，可以自定义一个列表拆分函数 list_split()。

In [9]	def list_split(content,sep):
	  new_list=[]
	  for i in range(len(content)):
	  new_list.append(list(filter(None,content[i].split(sep))))
	  return new_list
	list_split(S4,';;')

Out [9]	[['历史阐释', '历史事实', '历史评价', '唯物史观'], ['查理曼', '钦差巡察', '加洛林帝国', '法兰克', '中世纪'], ['南宋', '政治忌讳', '人物评价', '人际关系', '包容政治']]

6.2.2 字符串查询与替换

6.2.2.1 字符串查询

在 Python 中 in 可以实现直接查询(集合操作)。

In [10]	S5=['广州大学广州发展研究院','暨南大学文学院历史系','暨南大学管理学院'] '暨南大学' in S5[1]
Out [10]	True

根据 in 的特点可以自定义一个列表查询函数 find_words()。

In [11]	def find_words(content,pattern): return [content[i] for i in range(len(content)) if (pattern in content[i]) == True] find_words(S5,'暨南大学')
Out [11]	['暨南大学文学院历史系', '暨南大学管理学院']

同理，直接使用 len()函数就可以对所需查询内容的数量进行统计。

In [12]	len(find_words(S5,'暨南大学'))
Out [12]	2
In [13]	len(find_words(S5,'a'))
Out [13]	0

6.2.2.2 字符串替换

replace()函数可以对字符串的内容进行替换。

In [14]	'apple,orange'.replace("apple","banana")
Out [14]	'banana,orange'

可以自定义一个针对列表的字符串替换函数。

In [15]	def list_replace(content,old,new): return [content[i].replace(old,new) for i in range(len(content))] list_replace(S5,'暨南大学','华南农业大学')
Out [15]	['广州大学广州发展研究院', '华南农业大学文学院历史系', '华南农业大学管理学院']

6.3 网络爬虫及应用

网络爬虫又称网页蜘蛛或网络机器人，它按照一定的规则，自动抓取网络中的信息。它是一个自动提取网页的程序，为搜索引擎从互联网上下载网页，是搜索引擎的重要组成部分。下面介绍如何运用 Python 的 requests 和 bs4 两个第三方包将资料从网页中取出，导入 Python 中进行后续的处理。

在大数据时代，有相当多的资料都是通过网络来取得的，由于资料量日益增加，对于资料分析者而言，如何使用程序将网页中大量的资料自动汇入是很重要的事情。通过Python的网络爬虫技术，可以将大量结构化的资料直接导入Python中做数据分析，这样可以节省手动整理资料的时间。

6.3.1 网页的基础知识

6.3.1.1 网页资料结构

首先要简单介绍HTML的资料结构，以及CSS选择器(selector)的使用方式，有了这些观念才能精准地抓取网页中的资料。

目前网络上绝大部分网页都是以HTML格式来呈现的，因此若要抓取其中的资料，就必须对HTML的格式有初步的了解，这里简单介绍基本HTML的资料格式与概念，有了基本的概念才能做进一步的资料撷取。以下是一个简单的HTML网页原始程序代码。

```
<html>
  <head>
    <title>网页标题</title>
  </head>
  <body>
    <div class="container">
      <p>网页内容</p>
      <p>
        <ul> <li>foo</li> <li>bar</li> </ul>
      </p>
    </div>
  </body>
</html>
```

一个HTML网页中含有各种网页的元素(elements)，每个元素通常都会使用HTML的标签(tags)前后包起来，例如：

```
<p>网页内容</p>
```

而大部分HTML元素都是以巢状资料结构存在的，也就是说，一个元素中可能还会包含其他很多不同的元素，例如：

```
<ul>
    <li>foo</li>
    <li>bar</li>
</ul>
```

这种情况就是一个<ul元素中还包含两个<li元素。

基本上每个HTML网页中的资料都是以这样的阶层式规则呈现的，当要抓取网页中的资料时，只要明确得知资料在这个阶层结构中的位置，就可以很容易地将资料以编程方式自动抓取。若只是抓取网页资料，仅了解HTML基本的巢状结构概念即可，网页中的每个HTML标签都有不同的意义。

6.3.1.2 路径选择工具

SelectorGadget 是 Google Chrome 浏览器的一个外挂工具，可以用来显示网页中任意元素的 CSS 选择器路径，帮助快速撷取网页上的资料。有了 SelectorGadget，就可以直接定位所需要的数据，而不必学习复杂的网页设计等知识，结合 Python 就可以将需要的信息从网页中提取出来。有 SelectorGadget 和 Python 在手，没有任何计算机知识的人都可以轻松爬取网络数据。

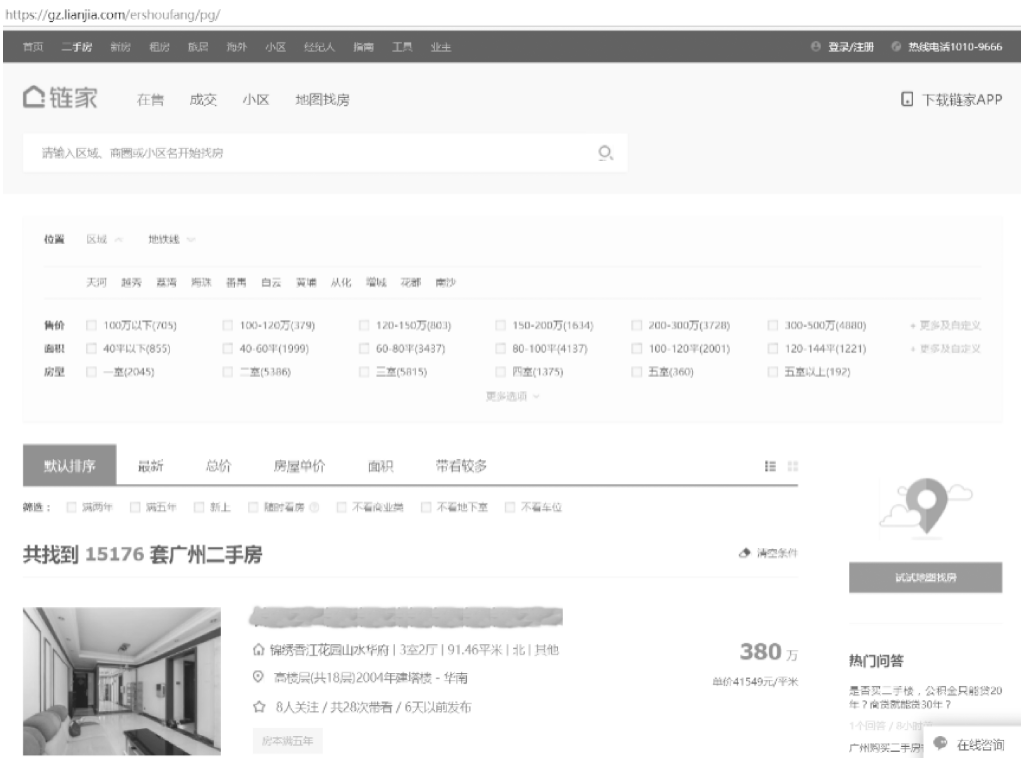
SelectorGadget 的安装方式有两种，一种是从 Chrome 在线应用程序商店直接安装（建议一般用户采用此方式），而另一种则是直接将 SelectorGadget 官方网站所提供的链接拖至浏览器书签，使用的时候单击该链接即可。

6.3.2 Python 爬虫步骤

6.3.2.1 读取网页

下面以链家网广州的二手房出售数据为例，系统地讲解数据爬虫的每个步骤。在谷歌浏览器中，同时按 Ctrl+U 键就可调出所要分析的源代码，网络爬虫实际上是利用网页的规则从网页源代码中检索出所需要的信息，因此本质就是一个文本搜索过程。

链家网址：https://gz.lianjia.com/ershoufang/pg



将 requests 和 bs4 中的函数整理成读取网页函数 read_html()，它可以将整个网页的原始 HTML 程序代码抓取下来。

In [16]	import requests def read_html(url,encoding='utf-8'):　　#定义读取 html 网页函数 　　headers={"User-Agent":"Mozilla/5.0（Macintosh; Intel Mac OS X 10_12_4） 　　AppleWebKit/537.36（KHTML, like Gecko）Chrome/58.0.3029.110 Safari/537.36" } 　　response =requests.get(url,headers=headers) 　　response.encoding = 'utf-8' 　　return(response.text)
In [17]	url='https://gz.lianjia.com/ershoufang'　#广州链家二手房网址 page=read_html(url)　　#读取网页 page
Out [17]	'<!DOCTYPE html><html><head><meta http-equiv="Content-Type" content="text/html; charset=utf-8"><meta http-equiv="X-UA-Compatible" content="IE=edge" /><meta http-equiv="Cache-Control" content="no-transform" /><meta http-equiv="Cache-Control" ..... <title>【广州二手房_广州二手房出售_广州二手房网】（广州链家网）</title>\n<meta name="description" content="链家广州二手房网,现有广州二手房真实房源 20308 套.为准备买广州二手房的用户提供广州地图找房、通勤找房等快捷找房工具,方便您更快捷的了解和购买广州二手房.买广州二手房就到广州链家网." /> .....

6.3.2.2　提取信息

选取网页节点的步骤如下：

① 打开链家页面，并开启 SelectorGadget 工具列，通常就会显示在页面的右下角。

② 鼠标点选要撷取的资料。被点选的 HTML 元素会以绿色标示，而这时 SelectorGadget 会尝试侦测用户要抓取资料的规则，产生一组 CSS 选择器并显示在 SelectorGadget 工具列上，同时，网页上所有符合这组 CSS 选择器的 HTML 元素都会以黄色标示，也就是说，目前这组 CSS 选择器会撷取所有绿色与黄色的 HTML 元素。

③ 通常 SelectorGadget 自动侦测的 CSS 选择器可能会包含一些我们不想要的资料，这时可用鼠标点选那些被标示为黄色但是应该排除的 HTML 元素。当鼠标点击黄色元素之后，该元素就会变成红色，并且将该元素排除在外。

④ 使用鼠标的选择与排除功能，将所有要撷取的元素精准地标示出来，产生一组精确的 CSS 选择器。有了这组精确的 CSS 选择器之后，就可以利用 Python 中的 info.select() 函数将资料直接截取至 Python 中处理了。

6.3.3　爬虫方法的应用

6.3.3.1　爬取链家二手房数据

In [18]	def html_text(info,word):　　#按关键词解析文本 　　return([w.get_text() for w in info.select(word)]) from bs4 import BeautifulSoup soup=BeautifulSoup(page,'lxml') houseInfo=html_text(soup,'.houseInfo'); houseInfo

Out [18]	['华南碧桂园叠翠苑 | 3 室 2 厅 | 106.2 平米 | 东北 | 其他 | 无电梯', '华景新城怡华台 | 3 室 2 厅 | 91 平米 | 西南 | 精装 | 有电梯', '骏景花园 | 3 室 2 厅 | 105 平米 | 南 北 | 简装 | 有电梯', '华南碧桂园紫翠苑 | 3 室 2 厅 | 107 平米 | 东 | 精装 | 无电梯', '百顺台花园 | 2 室 1 厅 | 75 平米 | 西 | 其他 | 有电梯', ...
In [19]	Price=html_text(soup,'.totalPrice span');Price
Out [19]	['318', '495', '500', '320', '260', '850', '280', '288', '235', '558', '310', '250', '279', '430', ...

6.3.3.2 批量下载二手房数据

前面的操作是针对某个网页的数据进行爬取。以广州链家网的二手数据为例，一共有 100 个网页的数据，如何将广州链家所有二手房的信息提取出来呢？只需总结这些网页的规律，使用循环函数（for()）重复上面的操作即可。

例如，从网址信息可以发现，第 1 页到第 2 页，第 2 页到第 3 页，变化的仅是末尾的序号。因此，在循环中可以将最后一位的数字以循环变量 i 替换。有时网页的序号出现在网址中间，有时出现在末尾。基本上所有的网络爬虫操作都需要总结网页的规律。

```
https://gz.lianjia.com/ershoufang/pg1
https://gz.lianjia.com/ershoufang/pg2
https://gz.lianjia.com/ershoufang/pg3
......
https://gz.lianjia.com/ershoufang/pgi
```

下面，爬取广州链家网所有二手房房价的数据，并提出一个统计分析思路：广州的二手房房价分布是否服从正态分布？要回答这个问题，可以爬取网站上所公布的全部二手房的房价数据并进行分析。因为所面对的数据不是事先准备好的数据集，而是直接从网络上爬取的第一手数据，因此对数据进行整理和清洗之后才可以进行数据分析。下面会介绍如何对该数据中出现的噪音进行清理，给读者提供一定的参考和借鉴。

可以将链家网所有有分析价值的信息（二手房的名称，二手房的描述，二手房的位置，二手房的整体房价和二手房的单位房价）全部爬取出来，自定义如下函数，然后写成 csv 或 xlsx 格式的文件，便于进一步分析。

针对单独的网页，可以通过数据框来存放网页的信息。

In [20]:
```
import PyDm_fun as dm
dm.lianjia_page(soup)
```

Out [20]:

	房屋信息	房屋价格	房屋位置	房屋单价
0	华南碧桂园叠翠苑|3 室 2 厅 |106.2 平米 |东北 |其他|无电梯	318	华南	单价 29944 元/平米
1	华景新城怡华台| 3 室 2 厅 | 91 平米 | 西南 | 精装 |有电梯	495	华景新城	单价 54396 元/平米
2	骏景花园 | 3 室 2 厅 | 105 平米 | 南 北 | 简装 | 有电梯	500	棠下	单价 47620 元/平米
3	华南碧桂园紫翠苑| 3 室 2 厅 | 107 平米 | 东 | 精装 |无电梯	320	华南	单价 29907 元/平米
4	百顺台花园 | 2 室 1 厅 | 75 平米 | 西 | 其他 | 有电梯	260	新市	单价 34667 元/平米

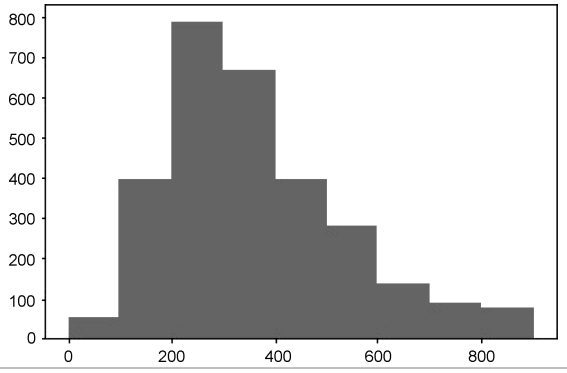
.....

可以测试一下爬取 10 个网页所花费的时间。（注：耗时与网速和电脑配置相关，因人而异。）

In [21]	#计算运行时间 %timeit LJdata=lianjia_all(url,10)
Out [21]	10.977s

6.3.3.3 爬虫数据的统计分析

In [22]	LJdata=pd.read_excel('PyDm_data.xlsx','LJdata') #读取所有广州链家数据
	Price=LJdata['房屋价格'];Price
	P=Price.astype(float);P
	P.describe()

Out [22]	count 3000.000000 mean 395.951567 std 259.472121 min 55.000000 25% 235.000000 50% 330.000000 75% 480.000000 max 3000.000000
In [23]	plt.hist(P,bins=np.arange(0,1000,100));
Out [23]	

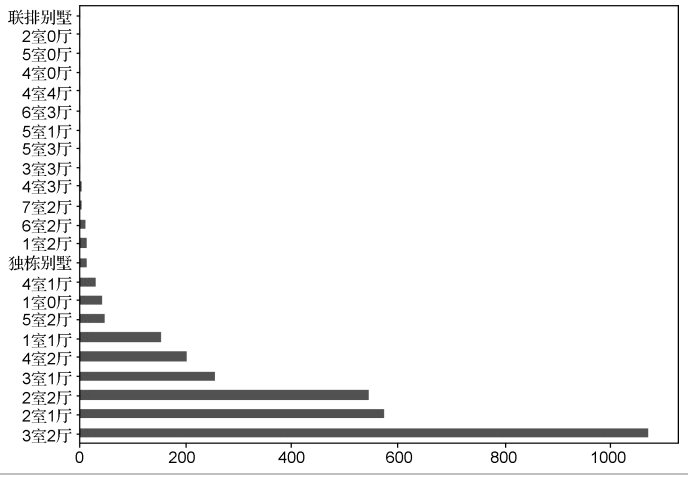
In [24]	House=LJdata['房屋信息']; #房屋信息提取
	House=[House[i].replace(' ','') for i in range(len(House))]
	House1=list_split(House,'|');House1[:6]

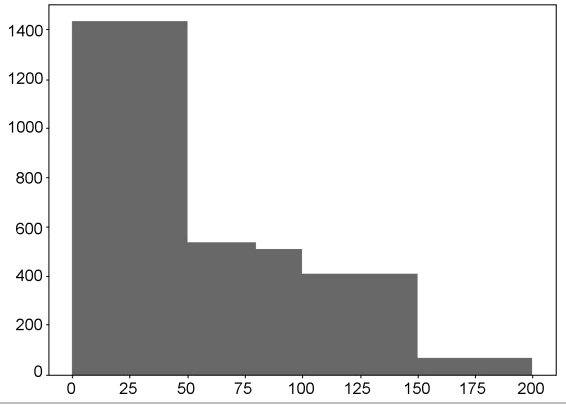
Out [24]	[['骏景花园', '3 室 2 厅', '105 平米', '南北', '简装', '有电梯'], ['华景新城怡华台', '3 室 2 厅', '91 平米', '西南', '精装', '有电梯'], ['芳草园', '3 室 2 厅', '117.05 平米', '北', '简装', '有电梯'], ['保利百合花园二期', '2 室 1 厅', '72.27 平米', '西南', '精装', '有电梯'], ['汇侨新城东区', '2 室 1 厅', '75.5 平米', '东北', '简装', '无电梯'], ['华南碧桂园紫翠苑', '3 室 2 厅', '107 平米', '东', '简装', '无电梯']] .....

In [25]	#数据清洗：去除第 7 位 NONA
	length=[len(House1[i]) for i in range(len(House1))];
	length_result=[idx for idx, e in enumerate(length) if e==7]
	for i in length_result: House1[i].remove('独栋别墅') #去除独栋别墅
	#去除在朝向中的不协调表述
	error_check=[i for i in range(len(House1)) if House1[i][1]=='联排别墅' or

	House1[i][1]=='独栋别墅'] for i in error_check: del House1[i][1] House1[-6:]
Out [25]	[['锦绣香江花园布查特国际公寓', '2 室 2 厅','126 平米','东南','其他','有电梯'], ['光大花园中海锦榕湾', '3 室 1 厅', '102 平米', '南', '其他', '有电梯'], ['东浚荔景苑', '3 室 1 厅', '95.97 平米', '东北', '简装'], ['广州雅居乐花园剑桥郡', '3 室 2 厅', '128.63 平米', '东南', '其他', '有电梯'], ['泓景花园', '3 室 2 厅', '110 平米', '东北', '其他', '有电梯'], ['下塘西路', '3 室 2 厅', '121.02 平米', '西北', '其他', '无电梯']] .....

In [26]	#构建分析用数据框
	House2=pd.DataFrame(House1)
	House2.info()
	House2.columns=['小区','格式','面积','朝向','装修','电梯']
	House2.head()

Out [26]	小区 格式 面积 朝向 装修 电梯 0 骏景花园 3 室 2 厅 105 平米 南北 简装 有电梯 1 华景新城怡华台 3 室 2 厅 91 平米 西南 精装 有电梯 2 芳草园 3 室 2 厅 117.05 平米 北 简装 有电梯 3 保利百合花园二期 2 室 1 厅 72.27 平米 西南 精装 有电梯 4 汇侨新城东区 2 室 1 厅 75.5 平米 东北 简装 无电梯 .....
In [27]	House2['小区'].value_counts()
Out [27]	骏景花园 41 富力桃园 40 翡翠绿洲森林半岛 31 碧桂园凤凰城凤馨苑 30 棠德花园 27 时代玫瑰园 22 汇侨新城北区 20 龙光峰景华庭 20 .....
In [28]	House2['格式'].value_counts().plot(kind='bar')
Out [28]	
In [29]	House2['面积'].value_counts()
Out [29]	80 平米 36

	95 平米 27 88 平米 26 77 平米 26 76 平米 25 97 平米 24 96 平米 24 .....
In [30]	House2['朝向'].value_counts()
Out [30]	南 728 北 587 南北 348 东南 325 东北 245 西南 224 东 200 西北 172 西 136 东西 34 暂无数据 1
In [31]	House2['装修'].value_counts()
Out [31]	其他 1451 简装 864 精装 631 毛坯 38
In [32]	House2['电梯'].value_counts()
Out [32]	有电梯 1646 无电梯 718
In [33]	import PyDm_fun as dm dm.freq(MJ,bins=[0,50,80,100,150,200])
Out [33]	[下限 上限) 频数 频率(%) 累计频数(%) 0 0.0 50.0 1433.0 48.38 48.38 1 50.0 80.0 540.0 18.23 66.61 2 80.0 100.0 510.0 17.22 83.83 3 100.0 150.0 410.0 13.84 97.67 4 150.0 200.0 69.0 2.33 100.00 

6.4 数据库技术及应用

前面讲到，大数据通常有 4V 特征，即体量大(Volume)、速度快(Velocity)、类型多(Variety)和价值大(Value)，但本书作为大数据分析的入门教程，不可能涉及大数据的方方面面，仅从大量数据分析出发，进行基本的大数据挖掘分析，牵扯到的也仅是传统的结构化数据，使用的也是过去的关系型数据库。

这类数据最典型的有人口普查数据、经济普查数据、金融证券数据、交通通信数据等，下面采用 Python 进行数据库数据的管理与分析，如果用不到数据库，可跳过本节内容。

6.4.1 Python 中数据库的使用

6.4.1.1 关系数据库的使用

当分析的数据量很大时，采用电子表格类软件有一大问题，即电子表格软件有数据限制，例如，Excel 2007 以下版本数据最大为 65560 条记录，虽然 Excel 2007 以上版本数据可包含百万级的数据行，但当数据超过几十万条以后，运行已很慢了，而且在 Excel 中直接分析这类数据已不现实。

Python 自身目前不易支持数据共享，因为当多个用户获取数据的时候，存在更新同一个数据的情况，这样，一个用户的操作对另外的用户就是不可见的了。

数据库管理系统，尤其是关系型数据库管理系统，可用来完成这些工作，其功能有如下方面：

① 提供读取大数据集中快速选取部分数据的功能；

② 数据库中强大功能的汇总和交叉列表的功能；

③ 以比长方形格子模型的电子表格更加严格的方式保存数据；

④ 多用户并发存取数据，同时确保存取数据的安全约束；

⑤ 作为一个服务器，为大范围的用户提供服务。

6.4.1.2 Python 中的数据库接口

网上有很多包可以实现 Python 和数据库的通信，它们提供了不同层次的抽象，有些提供了将整个数据框读入/写出到数据库的功能。这些包中都有通过 SQL 查询语言的函数来选取数据，选取的数据结果是分片的(通常是不同组的行)或者整体的(作为数据框)。

在 Python 中连接数据库需要安装其他扩展包，根据连接方式不同，我们有两种选择：一种是 ODBC(开放数据库接口)方式，需要安装 ODBC 驱动；另一种是基于 pandas 的 pandas.io.sql 模块的 SQLAlchemy 统一接口。SQLAlchemy 是 Python 编程语言下的一款 ORM 框架，该框架建立在数据库 API 之上，使用关系对象映射进行数据库操作，即将对象转换成 SQL，然后使用数据 API 执行 SQL 并获取执行结果。SQLAlchemy 的一个目标是提供能兼容众多数据库(如 SQLite、MySQL、PostgreSQL、Oracle、MSSQL、SQL Server 和 Firebird)的企业级持久性模型。

根据配置文件的不同调用不同的数据库 API，从而实现对数据库的操作，如：“数据库类型+数据库驱动名称://用户名:口令@机器地址:端口号/数据库名”。

```
from sqlalchemy import create_engine
MySQL:
   engine=create_engine('mysql+mysqldb://scott:tiger@localhost/foo')
MSSQL:
   engine=create_engine('mssql+pyodbc://mydsn')
Postgres:
   engine=create_engine('postgressql://scott:tiger@localhost:5432/mydatabase')
Oracle:
   engine=create_engine('oracle://scott:tiger@127.1.1.1:1521/sidname')
sqlite:
   engine=create_engine('sqlite:///foo.db')
```

这些数据库中 SQLite 是一个轻量级的数据库，完全免费，使用方便，不需要安装，无须任何配置，也不需要管理员。如果只需要本地单机操作，用它配合 Python 来存取数据是非常方便的。下面来看 Python 中操作 SQLite 数据库的示例。

6.4.2 数据库的建立与使用

6.4.2.1 sqlite 数据表的建立

从数据管理和编辑方便来说，最好的软件应该是微软的 Excel 和金山的 WPS 表格，大量的数据可以在一个电子表格工作簿中保存，但我们知道，电子表格对数据量是有数据限制的(Excel 2003 的最大行是 65536，从 Excel 2007 开始最大行是 1048576)。

对于文本和大量非结构化数据来说，使用电子表格保存和分析数据显然是不行的。当数据量很大时，通常需要用数据库来管理数据。最简单的数据库当属 SQLite3，大量的网站和小型研究通常是用 SQLite3 来管理数据的。

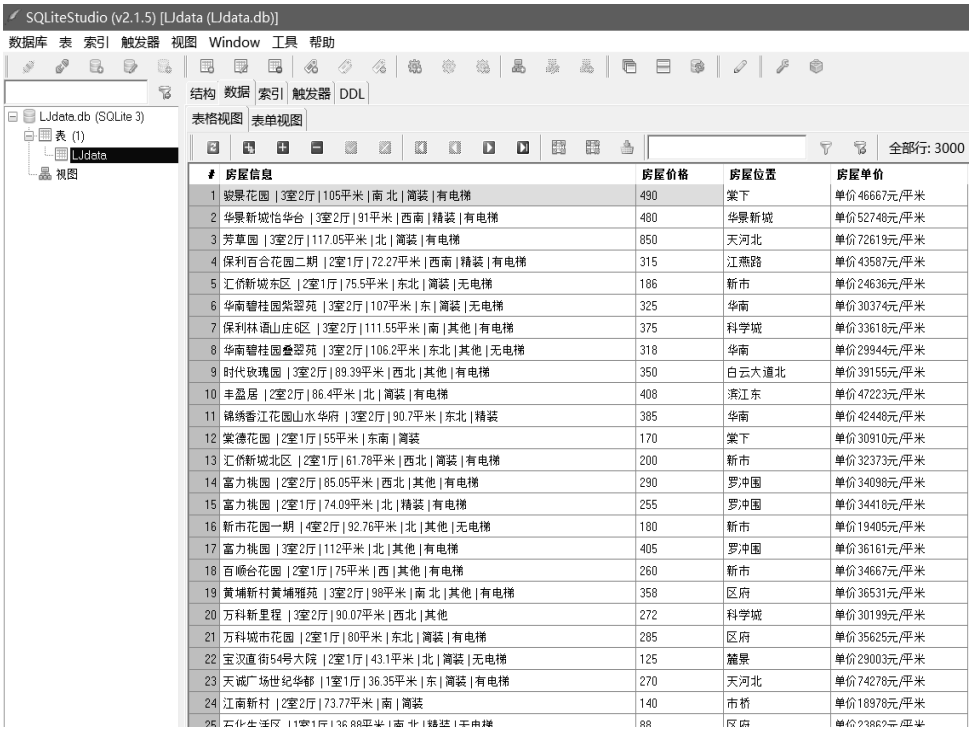
可以从网上下载一个 SQLite 可视化管理工具，对 SQLite3 数据库进行管理。SQLite Studio 是一款 SQLite 数据库可视化工具，是使用 SQLite 数据库开发应用的必备软件，软件无须安装，下载后解压即可使用，很小巧，但很实用，有绿色中文版本。比起其他 SQLite 管理工具，这个方便易用，不必安装单个可执行文件，支持中文。

下面的数据是从链家网站上爬取的有关二手房的数据(数据的具体爬取方法见 6.3 节)。

我们将爬取的文本数据存入 pandas 包的 SQLite3 数据库中：

In [34]	from sqlalchemy import create_engine engine=create_engine('sqlite:///LJdata.db') LJdata.to_sql('LJdata',engine,index=False)

然后在 SQLite Studio 中打开就能管理和编辑该数据库了。



#	房屋信息	房屋价格	房屋位置	房屋单价
1	骏景花园 | 3室2厅 | 105平米 | 南 北 | 简装 | 有电梯	490	棠下	单价46667元/平米
2	华景新城怡华台 | 3室2厅 | 91平米 | 西南 | 精装 | 有电梯	480	华景新城	单价52748元/平米
3	芳草园 | 3室2厅 | 117.05平米 | 北 | 简装 | 有电梯	850	天河北	单价72619元/平米
4	保利百合花园二期 | 2室1厅 | 72.27平米 | 西南 | 精装 | 有电梯	315	江燕路	单价43587元/平米
5	汇侨新城东区 | 2室1厅 | 75.5平米 | 东北 | 简装 | 无电梯	186	新市	单价24636元/平米
6	华南碧桂园紫翠苑 | 3室2厅 | 107平米 | 东 | 简装 | 无电梯	325	华南	单价30374元/平米
7	保利林语山庄6区 | 3室2厅 | 111.55平米 | 南 | 其他 | 有电梯	375	科学城	单价33618元/平米
8	华南碧桂园叠翠苑 | 3室2厅 | 106.2平米 | 东北 | 其他 | 无电梯	318	华南	单价29944元/平米
9	时代玫瑰园 | 3室2厅 | 89.39平米 | 西北 | 其他 | 有电梯	350	白云大道北	单价39155元/平米
10	丰盈居 | 2室2厅 | 86.4平米 | 北 | 简装 | 有电梯	408	滨江东	单价47223元/平米
11	锦绣香江花园山水华府 | 3室2厅 | 90.7平米 | 东北 | 精装	385	华南	单价42448元/平米
12	棠德花园 | 2室1厅 | 55平米 | 东南 | 简装	170	棠下	单价30910元/平米
13	汇侨新城北区 | 2室1厅 | 61.78平米 | 西北 | 简装 | 有电梯	200	新市	单价32373元/平米
14	富力桃园 | 2室2厅 | 85.05平米 | 西北 | 其他 | 有电梯	290	罗冲围	单价34098元/平米
15	富力桃园 | 2室1厅 | 74.09平米 | 北 | 精装 | 有电梯	255	罗冲围	单价34418元/平米
16	新市花园一期 | 4室2厅 | 92.76平米 | 北 | 其他 | 无电梯	180	新市	单价19405元/平米
17	富力桃园 | 3室2厅 | 112平米 | 北 | 其他 | 有电梯	405	罗冲围	单价36161元/平米
18	百顺台花园 | 2室1厅 | 75平米 | 西 | 其他 | 有电梯	260	新市	单价34667元/平米
19	黄埔新村黄埔雅苑 | 3室2厅 | 98平米 | 南 北 | 其他 | 有电梯	358	区府	单价36531元/平米
20	万科新里程 | 3室2厅 | 90.07平米 | 西北 | 其他	272	科学城	单价30199元/平米
21	万科城市花园 | 2室1厅 | 80平米 | 东北 | 简装 | 有电梯	285	区府	单价35625元/平米
22	宝汉直街54号大院 | 2室1厅 | 43.1平米 | 北 | 简装 | 无电梯	125	黄景	单价29003元/平米
23	天诚广场世纪华都 | 1室1厅 | 36.35平米 | 东 | 简装 | 有电梯	270	天河北	单价74278元/平米
24	江南新村 | 2室2厅 | 73.77平米 | 南 | 简装	140	市桥	单价18978元/平米
25	石化生活区 | 1室1厅 | 36.88平米 | 南 北 | 精装 | 无电梯	88	区府	单价23862元/平米

6.4.2.2 SQLite3 数据的处理

① 数据读取。

In [35]	from sqlalchemy import create_engine engine=create_engine('sqlite:///LJdata.db') LJ=pd.read_sql('LJdata',engine) LJ.info()
Out [35]	<class 'pandas.core.frame.DataFrame'> RangeIndex: 3000 entries, 0 to 2999 Data columns (total 4 columns): 房屋信息　　3000 non-null object 房屋价格　　3000 non-null float64 房屋位置　　3000 non-null object 房屋单价　　3000 non-null object dtypes: float64(1), object(3) memory usage: 93.8+ KB

② 其他分析从略，具体分析方法参见前面相关章节。

数据及练习 6

6.1 互联网电影资料库(Internet Movie Database，IMDb)是一个关于电影演员、电影、电视节目、电视明星和电影制作的在线数据库。IMDb 的资料中包括影片

的众多信息——演员、片长、内容简介、分级、评论等。对于电影的评分目前使用最多的就是 IMDb 评分。截至 2012 年 2 月 24 日，IMDb 共收录 2132383 部作品资料及 4530159 名人物资料。你可以尝试爬取其中感兴趣的信息，例如，爬取 2017 年度最流行的 100 部故事片，网址：http://www.imdb.com/search/title? %20count=100&release_date=2017, 2017&title_type=feature。请爬取以下信息：

Rank：从 1 到 100，代表排名；

Title：故事片的标题；

Description：电影内容简介；

Runtime：电影时长；

Genre：电影类型；

Rating：IMDb 提供的评级；

Metascore：IMDb 上该电影的评分；

Votes：电影的好评度；

Gross_Earning_in_Mil：电影总票房（百万）；

Director：影片的总导演，如果有多位，则取第一个；

Actor：影片的主演，如果有多位，则取第一个。

另外，还可以尝试爬取不同地区即将上映（upcoming releases）的电影名。

例如，尝试爬取中国的信息，网址：http://www.imdb.com/calendar?region=CN&ref_=rlm。

6.2 豆瓣读书。豆瓣读书为豆瓣网的一个子栏目。豆瓣读书 2005 年上线，已成为国内信息最全、用户量最大且最为活跃的读书网站。它专注于为用户提供全面且精细化的读书服务，同时不断探索新的产品模式。到 2012 年，豆瓣读书每月有 800 万以上名来访用户，过亿访问次数。

豆瓣用户每天都在对“读过”的书进行“很差”到“力荐”的评价，豆瓣根据每本书读过的人数以及该书所得的评价等综合数据，通过算法分析产生豆瓣图书 Top250。请尝试将读书榜 Top250 爬取下来。

网址：https://book.douban.com/top250?icn=index-book250-all。

6.3 百度新闻。百度新闻是百度公司推出的中文新闻搜索平台，每天发布多条新闻，新闻源包括 500 多个权威网站，热点新闻由新闻源网站和媒体每天“民主投票”选出，不含任何人工编辑成分，真实反映每时每刻的新闻热点；百度新闻保留了自建立以来所有日期的新闻，从而能掌握整个新闻事件的发展脉络。

尝试上百度新闻官网，爬取以“大数据”为关键词的全部新闻数据。

网址：http://news.baidu.com/ns?word=大数据&tn=news&from=news&cl=2&rn=20&ct=1。

6.4 BOSS 直聘。“BOSS 直聘”诞生于 2014 年 7 月，是一款让“牛人”和未来老板直接线上交流的 APP。用户可在 APP 上采用聊天的方式，与企业高管，

甚至创始人一对一沟通，更快地获得工作机会。“BOSS 直聘”为企业老板与职场“牛人”搭建起高效沟通、信息对等的公共平台。职场“牛人”可以跳过海投简历、一面、二面等冗长的应聘环节，直接与企业老板在线聊天、洽谈入职条件，提升找工作的效率。同时，企业老板也可采用类似微信聊天的在线互动方式，与求职者直接对话，展示自己和公司的诚意，精准定位职位最优人选，将招聘时长缩至最短。

尝试登录 BOSS 直聘官网，爬取广州地区所有职业招聘的数据。

网址：https://www.zhipin.com/c101280100/h_101280100/。

6.5 中原地产。香港中原集团始创于 1978 年，集团发展至今已成为香港规模最大的房地产代理集团。为发展及壮大中原集团，香港中原于 1990 年首度涉足中国大陆市场，并于 1992 年成立合资公司，1998 年成立独立运作的中原(中国)物业顾问有限公司。中原(中国)以为房地产公司提供专业化服务为依托，业务类型涉及房地产市场研究与分析、房地产前期顾问、房地产营销策划、广告设计、项目代理、物业管理、房产中介等。其业务范围还涉及投资移民、人事顾问、数据整合及软件开发等多个领域。

尝试登录中原地产官网，爬取广州地区所有二手房房价的数据。

网址：http://gz.centanet.com/ershoufang/。

第 7 章　文献计量与科研评价

文献计量学是指用数学和统计学的方法，定量地分析一切知识载体的交叉学科。它是集数学、统计学、文献学为一体，注重量化的综合性知识体系，其计量对象主要是文献量(各种出版物，尤以期刊论文和引文居多)、作者数(个人集体或团体)、词汇数(各种文献标识，其中以叙词居多),文献计量学最本质的特征在于其输出务必是“量”。

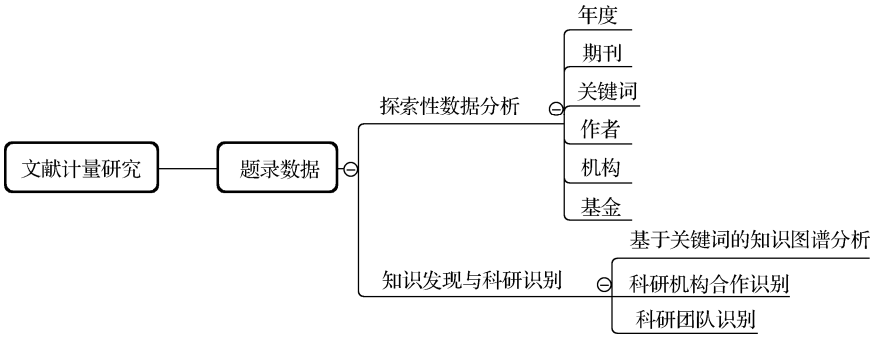
人们对文献定量化的研究，可以回溯到 20 世纪初。1917 年，F.J.科尔和 N.B.伊尔斯首先采用定量的方法，研究了 1543—1860 年所发表的比较解剖学文献，对有关图书和期刊文章进行统计，并按国别加以分类。1923 年，E.W.休姆提出“文献统计学”一词，并解释为，“通过对书面交流的统计及对其他方面的分析，以观察书面交流的过程，及某个学科的性质和发展方向。”1969 年，文献学家 A.普里查德提出用文献计量学代替文献统计学，他把文献统计学的研究对象由期刊扩展到所有书刊资料。目前，文献计量学已成为情报学和文献学的一个重要学科分支。同时也展现出重要的方法论价值，成为情报学的一个特殊研究方法。在情报学内部的逻辑结构中，文献计量学已渐居核心地位，是与科学传播及基础理论关系密切的学术环节。现在全世界每年发表的文献计量学学术论文约为 400～500 篇。

下面以管理学为例，介绍文献计量研究的实际应用。孙继伟和金晓玲(2014)以“问题管理”为检索词，对中国知网数据库检索到的 1984—2012 年发表的问题管理论文进行文献计量分析。他们发现，问题管理是一种来自实践、在实践驱动下发展起来的实践派理论，是跨学科、跨专业、应用广泛、特色鲜明的新兴理论。蒋建武和李南才(2015)从研究视角、研究内容和研究层次三个维度，对 1996 年以来发表在 CSSCI 来源期刊上的 197 篇有关临时雇佣论文进行了文献计量分析。魏峰等(2017)以 CSSCI 来源期刊为数据来源，收集了 1998—2015 年间与心理契约研究相关的 480 篇文章，并以之为研究对象进行文献计量分析，发现了国内心理契约研究存在的 6 大研究热点。这些文章的思路在于，确定某一研究话题(如“问题管理”“心理契约”等)，然后在中国知网数据库或中文社会科学引文索引数据库(CSSCI)收集相关论文，以这些论文为研究对象并通过文献计量研究已有的方法进行统计整理和知识发现。这些文章均发表于管理学核心期刊《管理学报》上，说明文献计量研究在科研识别与知识发现领域具有十分广阔的应用前景。

7.1　文献计量研究的框架

基本上，所有的文献计量分析都可以依据以下框架。首先，将题录数据导入 Python 之后，对数据进行探索性数据分析，根据需要统计某类主题出现的频数(如统计发表论文

数排名前十的作者等)。然后基于共现分析的思路构建共现矩阵，画知识图谱。根据这个研究框架，也可以写出高质量的文献计量研究的文章。



文献计量研究分析框架

文献计量分析的应用前景广阔。根据已发表的学术论文，可以总结出四大应用方向。

(1)对热门话题进行文献计量分析

当我们要对某一学术话题展开研究时，首先就要问：这一话题的研究进展如何？有谁在做？相关文献发表在哪些期刊上？通过文献计量分析就可以对这一话题的发文情况进行归纳和梳理。

可以登录中国知网选择包含这些主题词的文章，下载其文献题录数据进行分析，如云计算、3D 打印、人工智能、大数据、一带一路、社会网络分析、社区发现、聚类分析等。下面以“文献计量”为关键词，详细介绍如何进行文献计量分析。

(2)对某一期刊(或某类期刊)的发文数据进行总结评价

有时，我们需要对某一期刊的论文发表数据进行分析。例如，有不少研究者通过文献计量分析方法对情报学核心期刊《情报科学》的发文状况进行分析和解读，如：

孙仙阁. 2008 年《情报科学》文献计量分析.情报科学. 2009.

王敏.《情报科学》文献计量分析——以 2009 年为例.东北农业大学学报(社会科学版). 2011.

贾爱娟.基于文献计量的 2010 年《情报科学》分析解读.农业图书情报学刊. 2012.

肖荣荣.2003—2012 年《情报科学》文献计量分析.情报科学.2013.

牛浏.基于文献计量法的 2014 年《情报科学》分析.农业网络信息.2015.

(3)对某一科研单位若干年的数据总结评价(科学学与科研管理)

在高校或科研单位中，单位的科研成果管理十分重要。在本章的最后部分，介绍了中国农业科学院 2017 年中文论文的发表情况。

(4)中文数据(中国知网)和英文数据(Web of Science 数据库)的比较分析

在 Web of Science 数据库中也能导出文献题录数据，可以比较同一话题(如“大数据”等)中文论文与英文论文研究的异同。例如，王晰巍等 2013 年发表的文章“基于文献计量方法对中外信息生态学术论文比较研究”。

7.2 文献数据的获取与分析

7.2.1 文献数据的获取

7.2.1.1 文献数据的收集

下面介绍如何从中国知网(CNKI)获得文献题录数据，可以根据以下几个类型收集题录数据。

(1)基于主题的文献题录数据

(2)基于期刊的文献题录数据

(3)基于作者单位的文献题录数据

文献题录收集的步骤如下。

① 勾选或筛选所要分析的文献。

② 在中国知网网页右下角单击【导出参考文献】。

③ 在文献导出格式中选择【自定义】文献题录格式。

④ 勾选关键词、年、第一负责人和基金。

⑤ 选择 xls 格式。

7.2.1.2 文献数据的读取

以“文献计量”为篇名收集 2000—2017 年所发表的论文题录数据，以此数据集为例做文献计量分析。

In [1]	WXdata=pd.read_excel('PyDm_data.xlsx','WXdata'); WXdata.info()
Out [1]	<class 'pandas.core.frame.DataFrame'> RangeIndex: 831 entries, 0 to 830 Data columns (total 8 columns): Title 831 non-null object Author 828 non-null object

	Organ 831 non-null object Source 831 non-null object Keyword 831 non-null object FirstDuty 827 non-null object Fund 444 non-null object Year 831 non-null int64
In [2]	Wxdata.iloc[:,:4].head()
Out [2]	Title Author Organ Source 0 国际健康促进研究的演进脉络与前沿热点——基于 CiteSpace Ⅴ的文献计量与可视化分析 刘路; 史曙生; 南京师范大学体育科学学院; 盐城师范学院体育学院; 沈阳体育学院学报 1 国内区块链研究热点的文献计量分析 王发明; 朱美娟; 山东工商学院经济学院; 情报杂志 2 我国教育管理研究热点的追溯——基于《现代教育管理》(2009—2016 年)的文献计量和共词分析 丁亚东; 薛海平; 首都师范大学; 现代教育管理 3 我国民办高等教育研究十年回眸——基于文献计量与可视化分析 郭哲; 胡德鑫; 清华大学; 现代教育管理 4 从严治党研究的知识图谱、聚类维度与拓展空间——基于 Citespace 的文献计量分析 周鹏; 山东大学马克思主义学院; 洛阳理工学院马克思主义学院; 探索

从结果中可以看到，共有 831 篇文献纳入分析，有 8 个变量，其中 444 篇标注了基金。

7.2.2 文献数据的分析

7.2.2.1 科研单位与基金统计

首先通过 find_words()函数自定义一个搜索函数，它可以返回所有符合条件的值，然后通过 len()函数直接返回所有值的长度，这就是我们所要计数统计的结果，因此我们定义了科研单位与基金统计函数 search_university()。

In [3]	def find_words(content,pattern): #寻找关键词 return [content[i] for i in range(len(content)) if (pattern in content[i]) == True] def search_university(content,pattern): return len([find_words(content[i],pattern) for i in range(len(content)) if find_words(content[i],pattern) != []])

教育部官网提供了中国全部高等院校校名的信息，通过搜索函数可以对全国所有高校在这组数据中出现的频率进行统计。基金统计也是同样的原理，首先将从中国知网获取基金的名称整理成列表，通过搜索函数对所有基金出现的频率进行统计。

In [4]	university=pd.read_excel('PyDm_data.xlsx','university'); university.学校名称.head()
Out [4]	0 北京大学 1 中国人民大学

	2　　清华大学 3　北京交通大学 4　北京工业大学
In　[5]	fund=pd.read_excel('PyDm_data.xlsx','fund'); fund.基金名称.head()
Out [5]	0　　国家自然科学基金 1　国家高技术研究发展计划(863 计划) 2　基础研究重大项目前期研究专项 3　　国家科技支撑计划 4　国家重点实验室建设项目计划

将文献数据中的单位数据取出并分词。

In　[6]	def list_split(content,separator):　#分解信息 　　new_list=[] 　　for i in range(len(content)): 　　new_list.append(list(filter(None,content[i].split(separator)))) return new_list organ=list_split(WXdata['Organ'],';') len(organ)
Out [6]	831
In　[7]	organ[0:5]
Out [7]	[['南京师范大学体育科学学院', '盐城师范学院体育学院'], ['山东工商学院经济学院'], ['首都师范大学'], ['清华大学'], ['山东大学马克思主义学院', '洛阳理工学院马克思主义学院']]

从结果可以看到，武汉大学、吉林大学和南京大学发表文献计量分析论文的数量处于全国前列。

In　[8]	data1=pd.DataFrame([[i,search_university(organ,i)] for i in university['学校名称']]) data1.rename(columns={0:'学校名称',1:'频数'},inplace=True) data1.sort_values(by='频数',ascending = False)[0:10]
Out [8]	学校名称　频数 512　武汉大学　42 194　吉林大学　24 277　南京大学　24 268　上海大学　17 154　大连理工大学　17 58　中国科学院大学　15 430　山东大学　15 0　北京大学　14 105　中国人民武装警察部队学院　14 355　安徽大学　13

同理，也可以对基金的状况进行汇总统计和排名。

In [9]	jijin=list_split(WXdata['Fund'].dropna(axis=0,how='all').tolist(),';;') data2=pd.DataFrame([[i,search_university(jijin,i)] for i in fund['基金名称']]) data2.rename(columns={0:'学校名称',1:'频数'},inplace=True) data2.sort_values(by='频数',ascending = False)[0:10]
Out [9]	学校名称 频数 0 国家自然科学基金 91 5 国家社会科学基金 74 47 高等学校博士学科点专项科研基金 3 84 全国教育科学规划 3 3 国家科技支撑计划 2 46 国家留学基金 2 14 国家软科学研究计划 1

7.2.2.2 作者和关键词统计

词频分析法是利用能够表达文献核心内容的关键字在某一研究领域文献中出现的频次高低来确定该领域研究热点和发展动向的文献计量学方法。一篇论文的关键词是其研究内容的高度浓缩，某些关键词在其所在领域反复出现，可以反映这一领域的研究热点。这里根据文献题录数据的特点(作者用“;”分隔，关键词用“;;”分隔)，先将作者与关键词进行分词(list_split()函数)，然后直接汇总统计。

In [10]	keyword=list_split(WXdata['Keyword'].dropna(axis=0,how='all').tolist(),';;') keyword1=sum(keyword,[]) pd.DataFrame(keyword1)[0].value_counts()[:10]
Out [10]	文献计量 343 文献计量分析 133 文献计量学 91 知识图谱 52 统计分析 43 计量分析 40 共词分析 34 研究热点 34 CNKI 28 研究进展 23
In [11]	def list_replace(content,old,new): #清除信息中的空格 return [content[i].replace(old,new) for i in range(len(content))] author=list_replace(WXdata['Author'].dropna(axis=0,how='all').tolist(),',',';') author1=list_split(author,';') author2=sum(author1,[]) pd.DataFrame(author2)[0].value_counts()[:10]
Out [11]	鲍国海 11 邱均平 10 姜春林 8 黄鲁成 8 兰月新 8

	张志强 6 任增元 5 刘娅 5 沈君 5 蔡文伯 5

7.2.2.3 年份和期刊统计

年份数据和期刊数据结构比较简单，通过 value_counts()函数进行汇总统计即可。

In [12]	WXdata.Source.value_counts()[:10]
Out [12]	现代情报 75 情报科学 74 情报杂志 58 图书情报工作 50 中国科技期刊研究 37 科技管理研究 28 图书馆工作与研究 18 图书馆理论与实践 17 情报理论与实践 15 新世纪图书馆 15
In [13]	WXdata.Year.value_counts()
Out [13]	2017 105 2016 98 2014 96 2013 94 2015 92 2011 79 2012 71 2010 63 2009 37 2008 29 2007 19 2006 15 2005 12 2004 6 2001 5 2000 4 2003 3 2002 3

7.3 科研数据的管理与评价

探索性数据分析主要做的是简单的汇总统计，知识图谱和科学研究需要对数据进行进一步的加工，一般使用共现矩阵(又称耦合矩阵)，并在此基础上绘制知识图谱。共现

分析的原理主要是，对一组词(作者、机构等)两两统计它们在同一篇文献中出现的次数，以此为基础获得相应的共现矩阵。例如，作者 A 与作者 B 共合作了 20 篇文章，则他们共现矩阵的距离(或联系)就为 20。

7.3.1 科研单位与项目分析

在高校或科研单位中，单位的科研成果管理十分重要。中国知网数据库基本涵盖了所有中文论文发表的数据。通过前文所介绍的基于 Python 的文献计量分析的分析框架，可以流程化、模块化地对单位中文论文的发表情况进行汇总和梳理，满足单位科研统计的需要。下面以中国农业科学院为例，分析其院属单位 2017 年中文期刊的论文发表情况。这里将结果整理成报告用的表格形式，如表 7-1 和表 7-2 所示。

表 7-1 中国农业科学院院属单位 2017 年论文发表情况

院 属 单 位	频 数	院 属 单 位	频 数
农业资源与农业区划研究所	323	上海兽医研究所	103
果树研究所	310	草原研究所	101
农业信息研究所	253	油料作物研究所	88
北京畜牧兽医研究所	245	农业质量标准与检测技术研究所	81
农业经济与发展研究所	243	麻类研究所	78
作物科学研究所	221	农田灌溉研究所	72
植物保护研究所	219	生物技术研究所	64
农产品加工研究所	187	蜜蜂研究所	58
特产研究所	187	柑桔研究所	56
郑州果树研究所	173	蚕业研究所	43
棉花研究所	169	家禽研究所	32
农业环境与可持续发展研究所	163	甜菜研究所	31
蔬菜花卉研究所	147	水牛研究所	29
哈尔滨兽医研究所	140	农业部食物与营养发展研究所	19
兰州兽医研究所	128	水稻研究所	14
饲料研究所	127	甘薯研究所	11
烟草研究所	126	深圳农业基因组研究所	9
兰州畜牧与兽药研究所	124	环境保护科研监测所	8
茶叶研究所	113	南京农业机械化研究所	1

表 7-2 中国农业科学院院属单位 2017 年论文发表承担基金项目数

基 金 名	频 数
国家自然科学基金	1043
国家科技支撑计划	263

续表

基　金　名	频　　数
国家社会科学基金	45
国家重点基础研究发展计划(973 计划)	25
国家高技术研究发展计划(863 计划)	23
农业部软科学研究项目	12
农业部“948”项目	9
国家留学基金	5
国家星火计划	4
农业科技成果转化资金项目	3
高等学校博士学科点专项科研基金	2
攀登计划	1

(1)数据导入

In [14]	NKYWX=pd.read_excel('PyDm_data.xlsx','NKYWX'); NKYWX.shape NKYWX.iloc[:,:2].head()
Out [14]	(4368, 8) Title　　Author 0　乡村振兴的战略关键点及其路径　　刘合光; 1　转基因玉米和转基因大豆盲样检测方法　　李丽娜;金龙国;谢传晓;刘昌林; 2　桔梗的体外抗氧化活性及总多酚和黄酮苷元含量分析　朴向民;于营;Sin-Hee Han;Sang-Won Lee;王英平;郭靖; 3　未培养微生物分离培养技术研究进展　邢磊;赵圣国;郑楠;李松励;王加启; 4　食用植物油中转基因成分检测技术研究进展　李允静;肖芳;邵林;武玉花;万丹凤;吴刚;
In [15]	NKYDW=pd.read_excel('PyDm_data.xlsx','NKYDW'); NKYDW.head()
Out [15]	单位 0　作物科学研究所 1　植物保护研究所 2　蔬菜花卉研究所 3　农业环境与可持续发展研究所 4　北京畜牧兽医研究所

(2)单位统计与基金统计

中国农业科学院拥有 34 个直属研究所，统计这 34 个研究所的论文发表情况。

In [16]	organ=list_split(NKYWX['Organ'],';') data1=pd.DataFrame([[i,search_university(organ,i)] for i in NKYDW['单位']]) data1.rename(columns={0:'单位',1:'频数'},inplace=True) data1.sort_values(by='频数',ascending = False)[:8]
Out [16]	单位　频数 10　农业资源与农业区划研究所　323 20　果树研究所　310

	11 农业信息研究所 253 4 北京畜牧兽医研究所 245 9 农业经济与发展研究所 243 0 作物科学研究所 221 1 植物保护研究所 219 28 特产研究所 187
In [17]	jijin=list_split(NKYWX['Fund'].dropna(axis=0,how='all').tolist(),';;') data2=pd.DataFrame([[i,search_university(jijin,i)] for i in fund['基金名称']]) data2.rename(columns={0:'基金名称',1:'频数'},inplace=True) data2.sort_values(by='频数',ascending = False)[:12]
Out [17]	基金名称 频数 0 国家自然科学基金 1043 3 国家科技支撑计划 263 5 国家社会科学基金 45 6 国家重点基础研究发展计划(973 计划) 25 1 国家高技术研究发展计划(863 计划) 23 20 农业部软科学研究项目 12 72 农业部“948”项目 9 46 国家留学基金 5 26 国家星火计划 4 38 农业科技成果转化资金项目 3 47 高等学校博士学科点专项科研基金 2 7 攀登计划 1

7.3.2 科研期刊与作者分析

中国农业科学院高频作者、关键词和期刊统计的汇总统计如表 7-3、表 7-4 和表 7-5 所示。

In [18]	author=list_replace(NKYWX['Author'].dropna(axis=0,how='all').tolist(),',',';') author1=list_split(author,';') author2=sum(author1,[]) pd.DataFrame(author2)[0].value_counts()[:5]
Out [18]	刁其玉 49 李俊 26 张杰 25 王静 25 王磊 24

表 7-3 中国农业科学院论文发表数量排名前 30 位的作者

作 者	频 数	作 者	频 数	作 者	频 数
刁其玉	49	王海波	23	杨亚军	20
李俊	26	王孝娣	23	戴求仲	19
王静	25	郑楠	22	李鹏程	19
张杰	25	陈洪岩	21	史祥宾	19
刘凤之	24	李春义	21	童光志	19

续表

作　者	频　数	作　者	频　数	作　者	频　数
王加启	24	李光玉	21	王志强	19
王磊	24	刘崇怀	21	张宏福	19
王强	24	屠焰	21	张伟	19
曹卫东	23	毕金峰	20	李强	18
王凤忠	23	才学鹏	20	李亚兵	18

```
In [19]: keyword=list_split(NKYWX['Keyword'].dropna(axis=0,how='all').tolist(),';;')
         keyword1=sum(keyword,[])
         pd.DataFrame(keyword1)[0].value_counts()[:5]
Out[19]: 产量　92
         棉花　85
         玉米　56
         品质　55
         小麦　47
```

表 7-4　中国农业科学院排名前 30 位的关键词

关　键　词	频　数	关　键　词	频　数	关　键　词	频　数
产量	92	品种	39	马铃薯	27
棉花	85	研究进展	37	肉鸡	27
玉米	56	聚类分析	35	农艺性状	26
品质	55	烟草	35	紫花苜蓿	25
小麦	47	展望	35	中国	24
生长性能	45	烤烟	31	基因克隆	23
原核表达	45	甘蔗	30	梨	23
影响因素	41	遗传多样性	30	苜蓿	23
水稻	40	生产性能	29	苹果	23
奶牛	39	冬小麦	27	相关分析	23

```
In [20]: NKYWX.Source.value_counts()[:5]
Out[20]: 动物营养学报　116
         中国农业科学　112
         中国预防兽医学报　77
         中国畜牧兽医　76
         世界农业　74
```

表 7-5　中国农业科学院排名前 30 位的发文期刊

期　刊　名	频　数	期　刊　名	频　数
动物营养学报	116	植物遗传资源学报	49
中国农业科学	112	中国棉花	49
中国预防兽医学报	77	作物学报	47

续表

期　刊　名	频　　数	期　刊　名	频　　数
中国畜牧兽医	76	农业工程学报	46
世界农业	74	黑龙江畜牧兽医	40
农业展望	69	棉花学报	40
中国蔬菜	57	中国烟草科学	40
中国动物传染病学报	53	中国农业科技导报	39
中国兽医科学	52	中国食物与营养	39
植物保护	51	中国油料作物学报	38
畜牧兽医学报	50	生物技术通报	37
食品科学	50	中国果树	36
中国农学通报	50	安徽农业科学	35
果农之友	49	果树学报	34
园艺学报	49	农产品质量与安全	34

数据及练习 7

7.1　对热门话题进行文献计量分析。请以“问题管理”为检索词，对中国知网数据库检索到的 2000—2017 年发表的问题管理论文进行文献计量分析。内容和格式仿照：孙继伟,金晓玲.问题管理研究的文献计量分析[J].管理学报,2014,11(07):953-958.

7.2　对某一期刊的发文数据进行总结评价。请运用文献计量学的方法，对 2018 年出版的《情报科学》进行统计分析，并与《情报科学》近几年的有关统计数据对比。内容和格式仿照：孙仙阁.2008 年《情报科学》文献计量分析[J].情报科学, 2009,27(11): 1679-1683.

7.3　对某一科研单位若干年的数据总结评价。请以中国知网 CNKI 的中国学术文献网络出版总库作为数据统计源，检索 2002—2017 年安徽工程大学的学术文献，参照本章节的文献计量分析框架进行分析。内容和格式仿照：秦丽萍,桂云苗.基于 CNKI 的安徽工程大学学术文献计量分析[J].安徽工程大学学报,2013, 28(03):91-94.

7.4　对某一科研单位若干年的数据总结评价。请以五邑大学为例，对 CNKI 收录的五邑大学 2003—2017 年的科研论文，参照本章节的文献计量分析框架进行分析。内容和格式仿照：陈水生.地方高校科研论文文献计量分析——以五邑大学为例[J].高教论坛,2014(01):80-85.

7.5　对热门话题进行文献计量分析。请以“临时雇佣”为检索词，对中国知网数据库检索到的 2000 年以来发表的管理论文进行文献计量分析。内容和格式仿照：蒋建武,李南才.基于文献计量法的国内临时雇佣研究述评[J].管理学报,2015,12(04):619-624.

第 8 章　社会网络分析方法

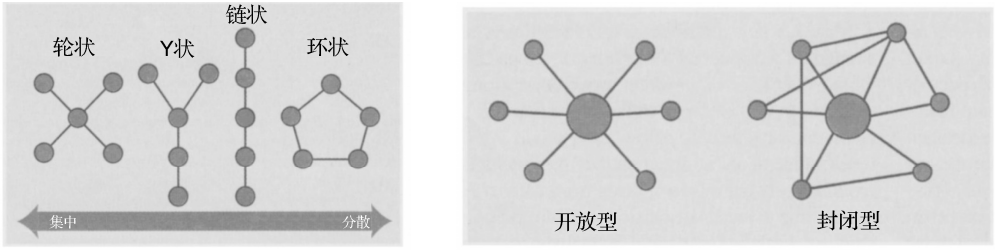
社会网络分析(social network analysis)最早是由社会学家根据图论和统计学知识发展起来的定量数据分析方法。近年来，该方法在经济学、农业科学、地理学、政治学等学科发挥了重要作用。学者们利用它可以得心应手地解释一些社会科学问题，是对社会网络的关系结构及属性加以分析的一套规范和方法。

8.1　社会网络的初步印象

8.1.1　社会网络分析概念

社会网络分析主要有两大要素：①行动者，在社会网络中用节点(node)表示；②关系，在社会网络中用连线(edge)表示，关系的内容可以是友谊、借贷或沟通，其关系可以是单向或双方的，且存在关系强度的差异，关系不同即构成不同的网络。社会学理论认为，社会不是由个人而是由网络构成的，网络中包含结点及结点之间的关系，社会网络分析法通过对网络中关系的分析，探讨网络的结构及属性特征，包括网络中的个体属性及网络整体属性，网络个体属性分析包括点度中心度、接近中心度等；网络的整体属性分析包括小世界效应、小团体研究、凝聚子群等。该方法目前在教育领域应用比较广泛，主要探究信息技术环境下学习者所构成网络的特点，以及在此基础上对于该网络的改进策略。

社会网络分析在学术领域得到广泛应用。世界顶级学术期刊《Science》于 2009 年专门刊登了题为《Network Analysis in the Social Sciences》的文章，详细介绍了如何用社会网络分析去解决实际问题。例如，社会网络图的结构类型如下。



社会网络图的结构类型

(左图上半部分别为轮状、Y 状、链状和环状，下半部为由集中到分散；右图为开放型和封闭型。)

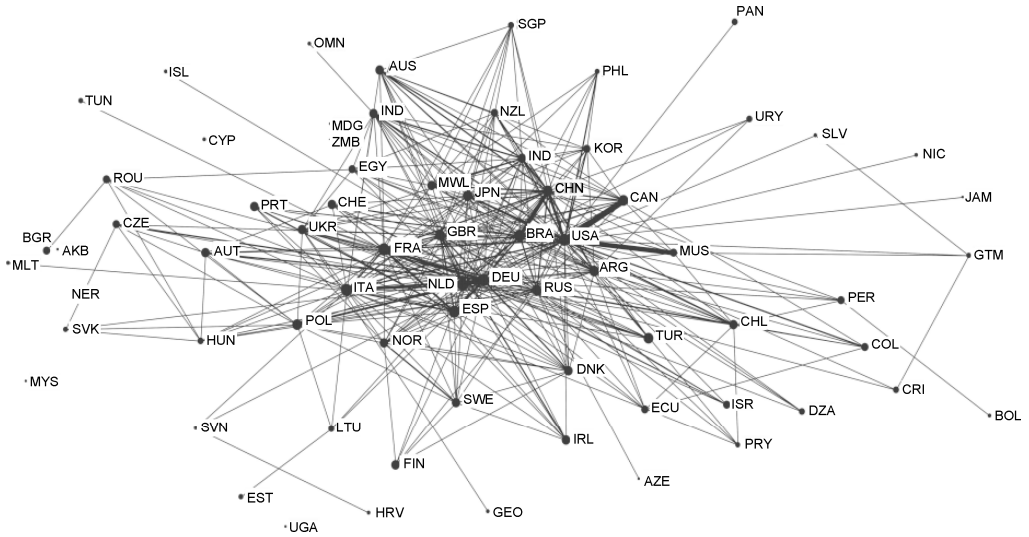
社会网络连线的不同属性如下(Borgatti S P, Mehra A, Brass D J, et al. Network Analysis in the Social Sciences[J]. Science, 2009, 323(5916):892-895.)。

Similarities			Social Relations				Interactions	Flows
Location e.g., Same spatial and temporal space	Membership e.g., Same clubs Same events etc.	Attribute e.g., Same gender Same attitude etc.	Kinship e.g., Mother of Sibling of	Other role e.g., Friend of Boss of Student of Competitor of	Affective e.g., Likes Hates etc.	Cognitive e.g., Knows Knows about Sees as happy etc.	e.g., Sex with Talked to Advice to Helped Harmed etc.	e.g., Information Beliefs Personnel Resources etc.

社会网络连线的不同属性

(第一栏为相似度，分别为位置(Location)、成员身份(Membership)、特征(Attribute)；第二栏为社会关系，分别为亲缘关系(Kinship)、其他社会关系(Other role)、影响力(Affective)、认识程度(Cognitive)；第三栏为交互关系；第四栏为网络流。)

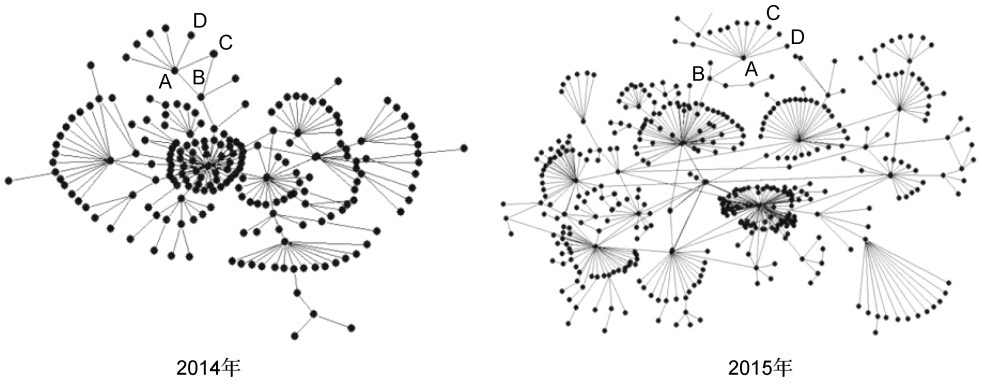
在经济管理研究领域，相较于最为常用的回归模型，社会网络分析能更好地分析和研究关系型数据。以管理学顶级杂志《管理世界》的刊文为例，马述忠等(2016)使用社会网络分析方法对农产品贸易网络特征及其对全球价值链分工的影响进行了研究。其中，节点代表不同的国家(或地区)，连线代表农产品贸易强度。(马述忠，任婉婉，吴国杰.一国农产品贸易网络特征及其对全球价值链分工的影响——基于社会网络分析视角[J].管理世界，2016(03):60-72.)



2013 年全球农产品贸易网络结构图

社会网络分析与其他统计建模方法结合能发挥更大的作用。刘善仕等(2017)基于在线简历数据构建人力资本社会网络，研究了人力资本社会网络与企业创新之间的关系。基于领英(中国)职业社交网站的人才简历数据，从社会网络视角构建了一个新颖的由人员流动形成的企业人力资本社会网络。

通过社会网络就可以得到新的变量(人力资本社会网络中心度等)，将这些变量作为自变量，与企业创新绩效(作为因变量)结合，通过回归分析，得出上市企业人力资本社会网络的中心度和结构与企业创新绩效呈显著正相关关系的结论。(刘善仕,孙博,葛淳棉,王琪.人力资本社会网络与企业创新——基于在线简历数据的实证研究[J].管理世界，2017(07):88-98+119+188.)



2014 年和 2015 年部分上市企业员工流动网络图

由这些案例可以发现，社会网络分析为学术研究提供了新的方法和思路，因此通过 Python 编程掌握这门技术是很有必要的。

8.1.2 社会网络分析包

networkx 是一个用 Python 语言开发的图论与复杂网络建模工具，内置了常用的图与复杂网络分析算法，可以方便地进行复杂网络数据分析和仿真建模等工作。networkx 支持创建简单无向图、有向图和多重图；内置许多标准的图论算法，节点可为任意数据；支持任意的边值维度，功能丰富，简单易用。这里只介绍网络建模的主要部分：构建网络和分析网络。

8.2 社会网络图的构建

8.2.1 社会网络数据形式

(1)以连线的形式构建网络

networkx 主要通过增加点和连线的方式构建网络。首先，通过 nx.Graph()函数创建一个空的网络。

In [1]	import networkx as nx nG=nx.Graph();nG
Out [1]	<networkx.classes.graph.Graph at 0x943de48>

可以选择一次增加一个节点函数 nG.add_node()，或用 nG.add_node_from()函数，它可以将任何可数的对象(如字段、列表和集合等)添加进网络。

In [2]	nG.add_node('JFK') nG.add_nodes_from(['SFO','LAX','ATL','FLO','DFW','HNL']) nG.number_of_nodes()
Out [2]	7

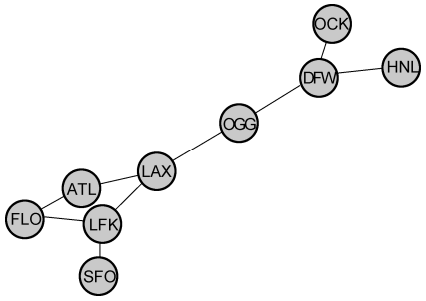
当网络中的节点确定时，可以添加连线。

In [3]	nG.add_edges_from([('JFK', 'SFO'), ('JFK', 'LAX'), ('LAX', 'ATL'), ('FLO','ATL'), ('ATL','JFK'), ('FLO','JFK'), ('DFW','HNL')]) nG.add_edges_from([('OKC','DFW'),('OGG','DFW'),('OGG','LAX')]) nG.number_of_edges()
Out [3]	10

nG.nodes()函数和 nG.edges()函数将返回网络中的节点和连线信息。

In [4]	nG.nodes() nG.edges()
Out [4]	NodeView(('JFK', 'SFO', 'LAX', 'ATL', 'FLO', 'DFW', 'HNL', 'OKC', 'OGG')) EdgeView([('JFK', 'SFO'), ('JFK', 'LAX'), ('JFK', 'ATL'), ('JFK', 'FLO'), ('LAX', 'ATL'), ('LAX', 'OGG'), ('ATL', 'FLO'), ('DFW', 'HNL'), ('DFW', 'OKC'), ('DFW', 'OGG')])

nx.draw()函数可以将网络数据可视化。

In [5]	nx.draw(nG, with_labels=True)
Out [5]	

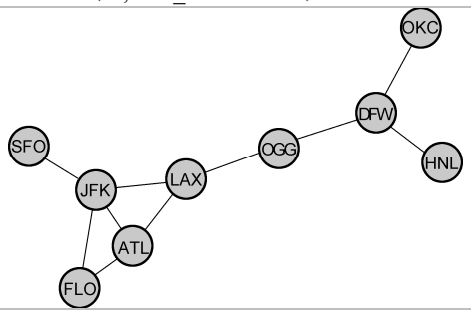
(2)以矩阵的形式构建网络

另一种常用的构建网络的方式是通过邻接矩阵的方式构建网络，如下面的矩阵所示。行列的交集代表两个点之间有无关系，0 代表无，1 代表有。

	JFK	SFO	LAX	ATL	FLO	DFW	HNL	OKC	OGG
JFK	0	1	1	1	1	0	0	0	0
SFO	1	0	0	0	0	0	0	0	0
LAX	1	0	0	1	0	0	0	0	1
ATL	1	0	1	0	1	0	0	0	0
FLO	1	0	0	1	0	0	0	0	0
DFW	0	0	0	0	0	0	1	1	1

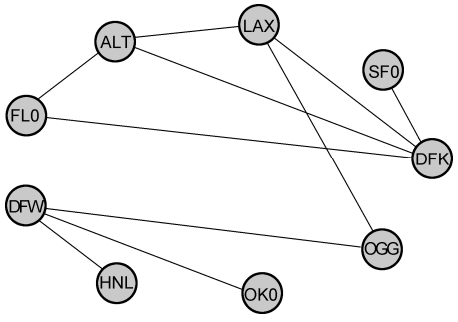
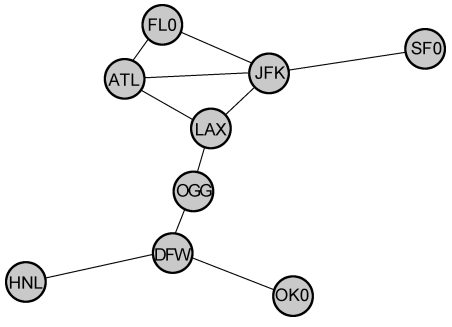
续表

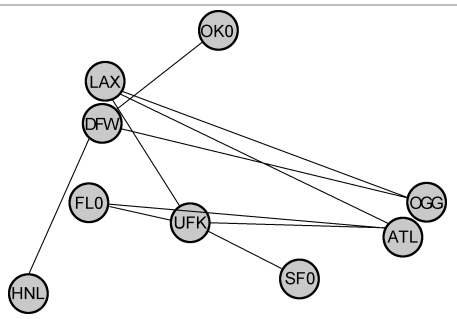
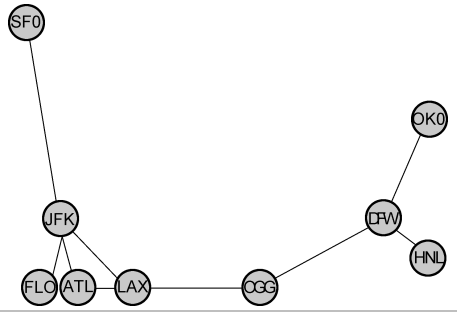
	JFK	SFO	LAX	ATL	FLO	DFW	HNL	OKC	OGG
HNL	0	0	0	0	0	1	0	0	0
OKC	0	0	0	0	0	1	0	0	0
OGG	0	0	1	0	0	1	0	0	0

In [6]	NXdata=pd.read_excel('PyDm_data.xlsx','NXdata',index_col=0) nf=nx.from_pandas_adjacency(NXdata) nx.draw(nf,with_labels=True)
Out [6]	

(3)社会网络图的布局

当网络的规模逐渐增大时，点与连线的布局也很重要。nx.draw()函数的“pos”参数提供了一系列网络布局的算法，可以对这些算法进行调试，选择令网络布局最美观的算法。

In [7]	nx.draw(nG,pos=nx.circular_layout(nG), with_labels=True)
Out [7]	
In [8]	nx.draw(nG,pos=nx.kamada_kawai_layout(nG), with_labels=True)
Out [8]	

In [9]	nx.draw(nG,pos=nx.random_layout(nG), with_labels=True)
Out [9]	
In [10]	nx.draw(nG,pos=nx.spectral_layout(nG), with_labels=True)
Out [10]	

8.2.2 社会网络统计量

社会网络分析常用的统计量如下。

(1) 网络汇总信息

提取网络汇总信息的函数如下：

In [11]	nx.info(nG)
Out [11]	'Name: \nType: Graph\nNumber of nodes: 9\nNumber of edges: 10\n Average degree: 2.2222'

(2) 网络密度

网络密度可用于刻画网络中节点间相互连边的密集程度，定义为网络中实际存在的边数与可容纳的边数上限的比值。在线社交网络中常用来测量社交关系的密集程度及演化趋势。一个具有 N 个节点和 L 条实际连边的网络，其网络密度为

$$d(G)=\frac{2L}{N(N-1)}$$

网络密度取值范围为[0，1]，当网络为全连通时，$d(G)$=1；当网络中不存在连边关系时，$d(G)$=0。

In [12]	nx.density(nG)
Out [12]	0.2777777777777778

(3) 网络直径

网络直径是指网络中任意两节点间距离的最大值，一般用链路数来度量。在社会网

络中，有时会出现几个互相不存在联系的成分（component），此时如果直接计算网络直径，那么理论上该网络的直径为无穷大。因此，当网络中出现多个成分的情况时，往往只计算规模最大的成分的直径。

In [13]	nx.diameter（nG）
Out [13]	5

该网络的直径为 5 步，说明在这个网络中最疏远的节点通过 5 个节点就可以产生联系，六度分离理论就是这样计算出来的。

六度分离（六度区隔）理论：你和任何一个陌生人之间所间隔的人不会超过 5 个，也就是说，最多通过 5 个人你就能够认识任何一个陌生人。根据这一理论，你和世界上任何一个人之间只隔着 5 个人，不管对方在哪个国家，属于哪类人种，是哪种肤色。

（4）聚类系数与相邻节点

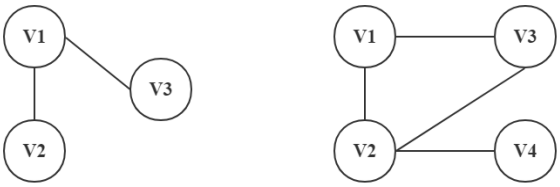
聚类系数是表示一个图形中节点聚集程度的系数，资料显示，在现实的网络中，尤其是在特定的网络中，由于相对高密度连接点的关系，节点总是趋向于建立一组严密的组织关系。在现实世界的网络中，这种可能性往往比两个节点之间随机建立一个连接的平均概率更大。这种相互关系可以利用聚类系数进行量化表示。

在很多网络中，如果节点 v1 连接于节点 v2，节点 v2 连接于节点 v3，那么节点 v3 很可能与 v1 相连接。这种现象体现了部分节点间存在的密集连接性质。例如，在无向网络中，可以用系数 CC（Cluster Coefficient）来表示 v2 的聚类系数：

$$\mathrm{CC}_{v2}=\frac{n}{\mathrm{C}_k^2}=\frac{2n}{k(k-1)}$$

式中，k 表示节点 v2 的所有相邻节点的个数，即节点 v2 的邻居；

n 表示节点 v2 的所有相邻节点之间相互连接的边的个数。



通过一个例子可以更好地理解如何计算聚类系数。上图中左边的节点中，v2 的聚类系数为 2/2（2−1）=1，而右边 v2 节点的聚类系数为 2/3（3−1）=2/3。

In [14]	nx.clustering（nG）
Out [14]	{'ATL': 0.6666666666666666, 'DFW': 0, 'FLO': 1.0, 'HNL': 0, 'JFK': 0.3333333333333333, 'LAX': 0.3333333333333333, 'OGG': 0, 'OKC': 0, 'SFO': 0}

网络的传递性(transitivity)是表示一个图形中节点聚集程度的系数，一个网络有一个值，可以衡量网络的关联性，值越大，表示交互关系越大，网络越复杂。

In [15]	nx.transitivity(nG)
Out [15]	0.35294117647058826

neighbors()函数可以返回某点所有相邻节点的信息。

In [16]	list(nG.neighbors('ATL'))
Out [16]	['LAX', 'FLO', 'JFK']

(5)中心性

度中心性(degree centrality)是在网络分析中刻画节点中心性(centrality)的最直接度量指标。一个节点的度越大，意味着这个节点的度中心性越强，该节点在网络中越重要。

In [17]	nx.degree_centrality(nG)
Out [17]	{'ATL': 0.375, 'DFW': 0.375, 'FLO': 0.25, 'HNL': 0.125, 'JFK': 0.5, 'LAX': 0.375, 'OGG': 0.25, 'OKC': 0.125, 'SFO': 0.125}

接近中心性(closeness centrality)反映在网络中某一节点与其他节点之间的接近程度。将一个节点到其他所有节点的最短距离累加起来，其倒数表示接近中心性。即对于一个节点，它距离其他节点越近，那么它的接近中心性越强。接近中心性需要考量每个节点到其他节点的最短路径的平均长度。也就是说，对于一个结点而言，它距离其他结点越近，那么它的中心度越高。一般来说，那种需要让尽可能多的人使用的设施，其接近中心度是比较高的。

In [18]	nx.closeness_centrality(nG)
Out [18]	{'ATL': 0.4444444444444444, 'DFW': 0.42105263157894735, 'FLO': 0.34782608695652173, 'HNL': 0.3076923076923077, 'JFK': 0.47058823529411764, 'LAX': 0.5333333333333333, 'OGG': 0.5, 'OKC': 0.3076923076923077, 'SFO': 0.3333333333333333}

中介(中间)中心性(between centrality)是以经过某个节点的最短路径数目来刻画节

点重要性的指标，指一个结点充当其他两个结点之间最短路径的“桥梁”的次数。一个结点充当“中介”的次数越多，它的中介中心度就越大。

In [19]	nx.betweenness_centrality(nG)
Out [19]	{'ATL': 0.08928571428571427, 'DFW': 0.46428571428571425, 'FLO': 0.0, 'HNL': 0.0, 'JFK': 0.33928571428571425, 'LAX': 0.5714285714285714, 'OGG': 0.5357142857142857, 'OKC': 0.0, 'SFO': 0.0}

(6)最短路径计算

计算最短路径的方法及 Python 实现如下：

In [20]	nx.shortest_path(nG,'ATL','SFO')
Out [20]	['ATL', 'JFK', 'SFO']
In [21]	len(nx.shortest_path(nG,'ATL','SFO'))
Out [21]	3

选好了起点和终点，shortest_path()函数可以在网络中寻找并返回最短距离的路径。直接用 len()函数就可以得到路径的距离。

8.2.3 网络图之知识图谱

知识图谱，又称为科学知识图谱，在图书情报界称为知识域可视化或知识领域映射地图，是显示知识发展进程与结构关系的一系列不同的图形，用可视化技术描述知识资源及其载体，挖掘、分析、构建、绘制和显示知识及它们之间的相互联系。可以将文献题录数据通过共现分析获得共现矩阵，然后通过前文介绍的社会网络分析方法，用可视化的图谱形象地展示学科的核心结构、发展历史、前沿领域，为学科研究提供切实的、有价值的参考。

8.2.3.1 图谱共现矩阵

下面对第 7 章中的文献数据，分别构造作者的共现矩阵、机构的共现矩阵和关键词的共现矩阵，如表 8-1、表 8-2 和表 8-3 所示，这些矩阵是做知识图谱和科研发现的基础。

In [22]	##提取第 7 章文献数据的高频数据 organ=list_split(WXdata['Organ'],';') data1=pd.DataFrame([[i,search_university(organ,i)] for i in university['学校名称']]) data1.rename(columns={0:'学校名称',1:'频数'},inplace=True) keyword=list_split(WXdata['Keyword'].dropna(axis=0,how='all').tolist(),';;') keyword1=sum(keyword,[])

	author=list_replace(WXdata['Author'].dropna(axis=0,how='all').tolist(),',',';') author1=list_split(author,';') author2=sum(author1,[])

In [23]	#获取前 30 名的高频数据
	data_author=pd.DataFrame(author2)[0].value_counts()[:30].index.tolist()
	data_keyword=pd.DataFrame(keyword1)[0].value_counts()[0:30].index.tolist()
	data_university=data1.sort_values(by='频数',ascending = False)[0:30]['学校名称'].tolist()

In [24]	def occurence(data,document): #定义共现矩阵
	empty1=[];empty2=[];empty3=[]
	for a in data:
	for b in data:
	count = 0
	for x in document:
	if [a in i for i in x].count(True) >0 and [b in i for i in x].count(True) >0:
	count += 1
	empty1.append(a);empty2.append(b);empty3.append(count)
	df=pd.DataFrame({'from':empty1,'to':empty2,'weight':empty3})
	G=nx.from_pandas_edgelist(df, 'from', 'to', 'weight')
	return(nx.to_pandas_adjacency(G, dtype=int))

In [25]	Matrix1=occurence(data_author,author);Matrix1

表 8-1 高频作者共现矩阵

	鲍国海	邱均平	黄鲁成	姜春林	兰月新	. . .
鲍国海	11	0	0	0	0	...
邱均平	0	10	0	0	0	...
黄鲁成	0	0	8	0	0	...
姜春林	0	0	0	8	0	...
兰月新	0	0	0	0	8	...
. . .	...	...	...	...	...	...

In [26]	Matrix2=occurence(data_university,organ)

表 8-2 高频发文单位共现矩阵

	武汉大学	吉林大学	南京大学	大连理工大学	上海大学	. . .
武汉大学	42	0	0	0	0	...
吉林大学	0	24	0	1	0	...
南京大学	0	0	24	0	0	...
大连理工大学	0	1	0	17	0	...
上海大学	0	0	0	0	17	...
. . .	...	...	...	...	...	...

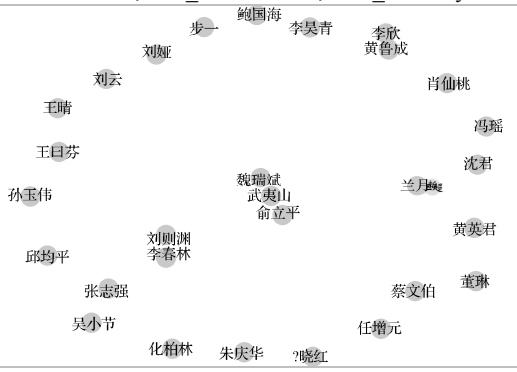
In [27]	Matrix3=occurence(data_keyword,keyword)

表 8-3　高频关键词共现矩阵

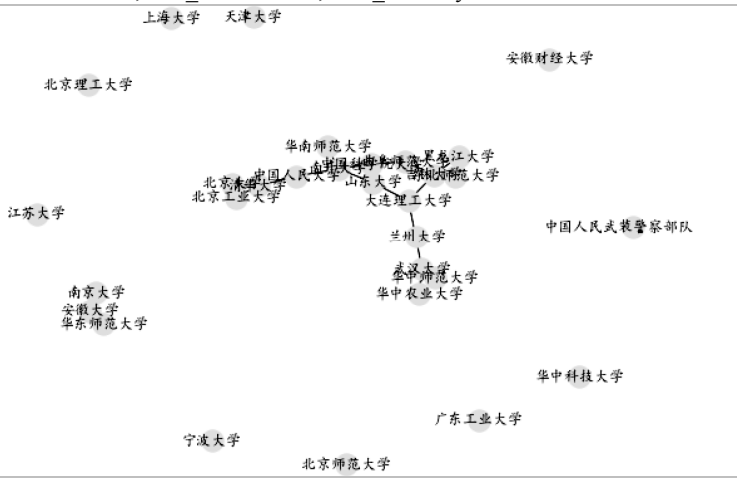
	文献计量	文献计量分析	文献计量学	知识图谱	统计分析	. . .
文献计量	642	138	94	43	31	...
文献计量分析	138	138	0	8	8	...
文献计量学	94	0	94	4	7	...
知识图谱	43	8	4	60	0	...
统计分析	31	8	7	0	43	...
. . .	...	...	...	...	...	...

8.2.3.2　共现矩阵网络图

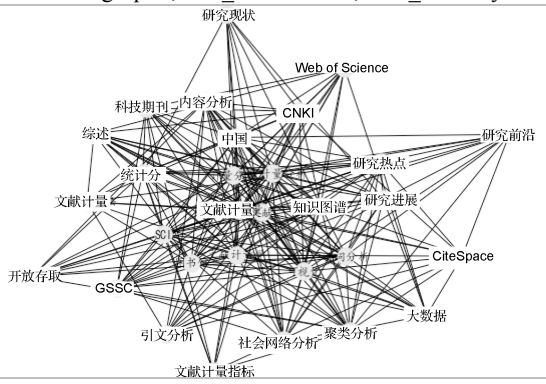
将共现矩阵转化为网络数据（用 nx.from_pandas_adjacency() 函数），然后用 nx.draw() 函数就可以画出高频作者间的合作情况。

In　[28]	import networkx as nx nf1=nx.from_pandas_adjacency（Matrix1） nx.draw（nf1,with_labels=True,node_color='yellow'）
Out [28]	

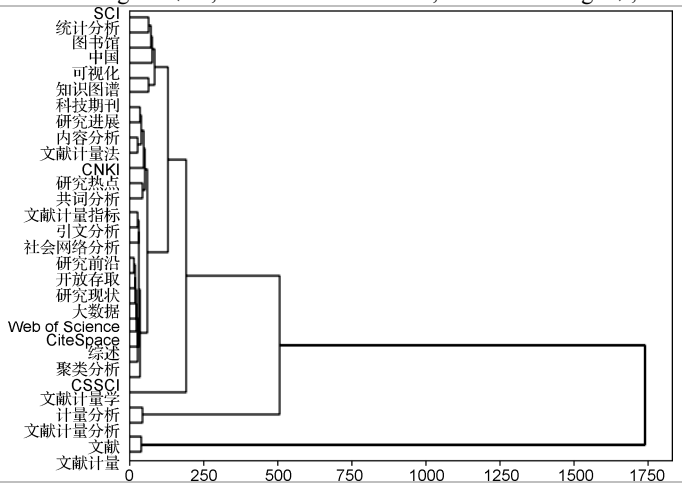
类似地，可以画出高校间合作的知识图谱。

In　[29]	nf2=nx.from_pandas_adjacency（Matrix2） nx.draw（nf2,with_labels=True,node_color='yellow'）
Out [29]	

下面是关键词间的知识图谱。

In [30]	graph3=nx.from_pandas_adjacency(Matrix3) nx.draw(graph3,with_labels=True,node_color='yellow')
Out [30]	

关键词的网络图往往不易区分出类别，这里倾向于画系统聚类图。通过系统聚类分析，可以将共现频率高的关键词找出来，从而总结与发现文献计量分析的研究状况与趋势。

In [31]	import scipy.cluster.hierarchy as sch H1=sch.linkage(Matrix3,method='ward'); sch.dendrogram(H1,labels=Matrix3.index,orientation='right');
Out [31]	

在画系统聚类图时，定义类与类之间的距离有许多种方法(如最短距离法、Ward 法等)，可以根据需要和实际效果选择最佳的聚类方法。

数据及练习 8

8.1 Bali 数据集[①]包含了特殊人群之间的关系数据，该数据是网络数据格式，由 17 个节点和 126 条连线构成，请画其网络图。

① 数据来自 R 语言 UserNetR 包。

8.2 DHHS 数据集[①]包含了用于评估美国烟草公司的领导力和控制力的合作网络数据。该数据是网络数据格式，由 54 个点和 477 条连线构成，请画其网络图。

8.3 FIFA_Nether 数据集[②]包含了 2010 年世界杯荷兰球员之间的传球数据，该数据是网络数据格式，由 11 个节点和 108 条连线构成，请画其网络图。

8.4 ICTS_G10 数据集[③]包含了华盛顿大学临床和转化科学学院科学家之间的合作数据，该数据是网络数据格式，由 493 个节点和 1359 条连线构成，连线的含义是，若两个科学家共同在一个资助项目工作，则代表他们之间存在联系，请画其网络图。

8.5 Krebs 数据集[④]包含了特殊人群之间的关系数据，该数据是网络数据格式，由 19 个节点和 27 条连线构成，请画其网络图。

①②③④ 数据来自 R 语言 UserNetR 包。

第 9 章　数据分析编程平台

如果用来讲课或演示数据分析结果，则推荐 Jupyter Notebook 平台，它有类似于 Mathematica 的界面，特点是可同时查看代码和运行结果，支持多种语言功能。如果用来做数据分析，建议用 Spyder 平台；如果用来做大工程，可考虑使用其他开发环境，如 Pycharm 等。你会发现，Matlab，Rstudio，Spyder 三者“长得”很像，说明做数据分析就应该是这样的界面。一个用熟了，其他两个就很容易上手了，可以将三者的常用功能的快捷键改成一致。

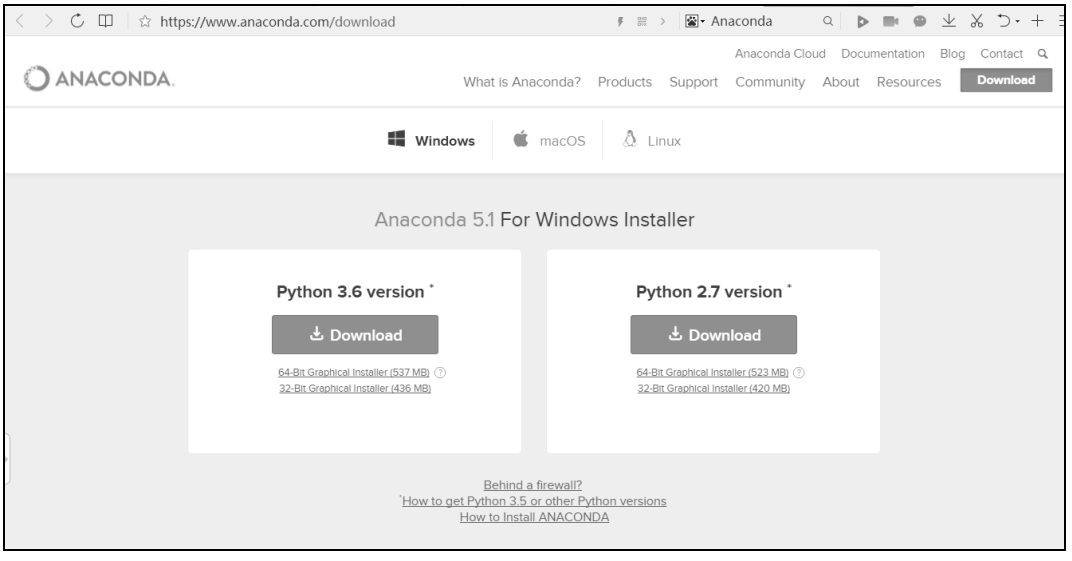
9.1　Anaconda 科学计算发行包

9.1.1　Anaconda 下载与安装

我们知道，基本的 Python 环境只包含基本的编程模块，基本不包含数据分析和科学计算模块，所以作为数据分析工作者，我们需要选择一个方便的 Python 编程环境。

可喜的是，现在有许多公司为了迎接大数据时代的来临，构建了许多基于 Python 的发行版，其中包含用于编程的 IDE（Integrated Development Environment，集成开发环境）、常用的编程和数据分析包。

这里给大家推荐一款用于科学计算和数据分析的 Python 的发行版 Anaconda，可登录 https://www.anaconda.com/下载其安装包，推荐 Python 3.6 及以上版本。

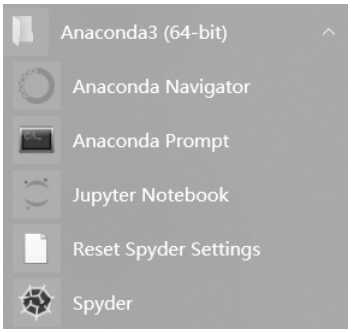


注意：Anaconda 指的是一个开源的 Python 发行版本，包含 numpy、pandas、matplotlib、scipy 等 180 多个科学包及其依赖项。因为包含大量的科学计算包，故 Anaconda 的下载文件比较大（约 500 MB），但安装后可满足大多数数据分析的需求。

下载 Windows 版 Anaconda 的 Python 3.6 版本，按常规方法安装，安装后在 Windows 系统菜单中会出现子菜单，可选择其中一个程序来使用 Python。

9.1.2 Anaconda 启动与运行

在 Windows 中安装好 Anaconda 后，将会在 Windows 菜单中出现下面的界面。



Anaconda 菜单：

系统安装导航

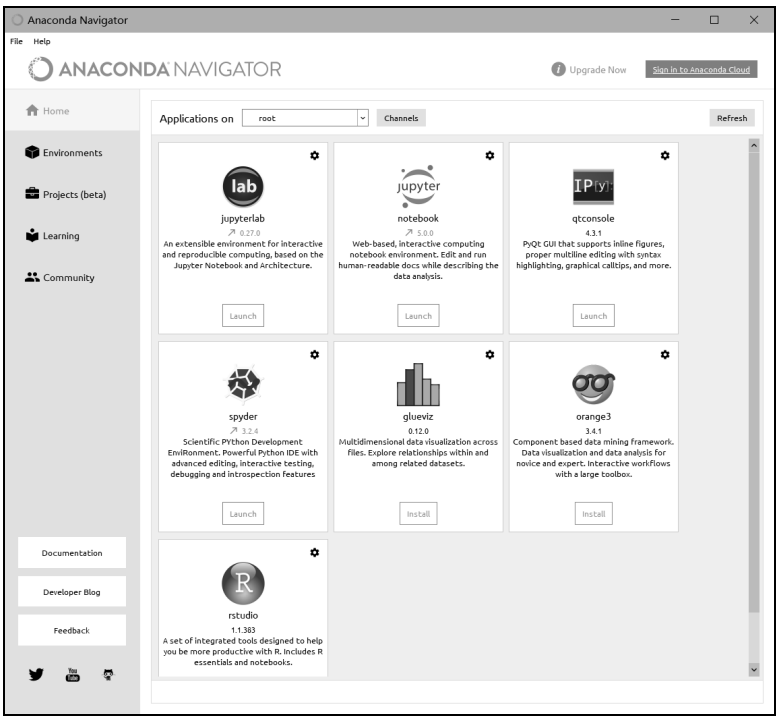
命令行系统

数据分析、展示系统

数据分析与开发平台

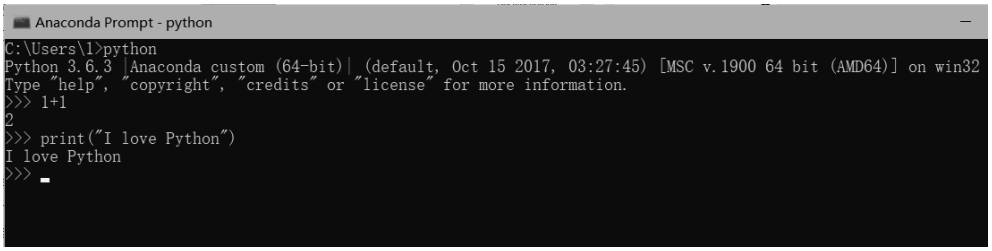
① 系统安装导航。

单击 Anaconda Navigator 进入下面的界面。



该导航系统里有大量的学习材料和平台，大家可选择使用和学习。

② 如果只作为计算器使用或进行简单计算，可直接执行 Anaconda Prompt，相当于在命令行执行 Python。



通常第三方程序包在此安装，安装命令为 pip install 包名或 conda install 包名，如要安装 Nbextensions 扩展包，则在命令行执行

```
>>> pip install jupyter_contrib_nbextensions
```

下面是一些包的命令。

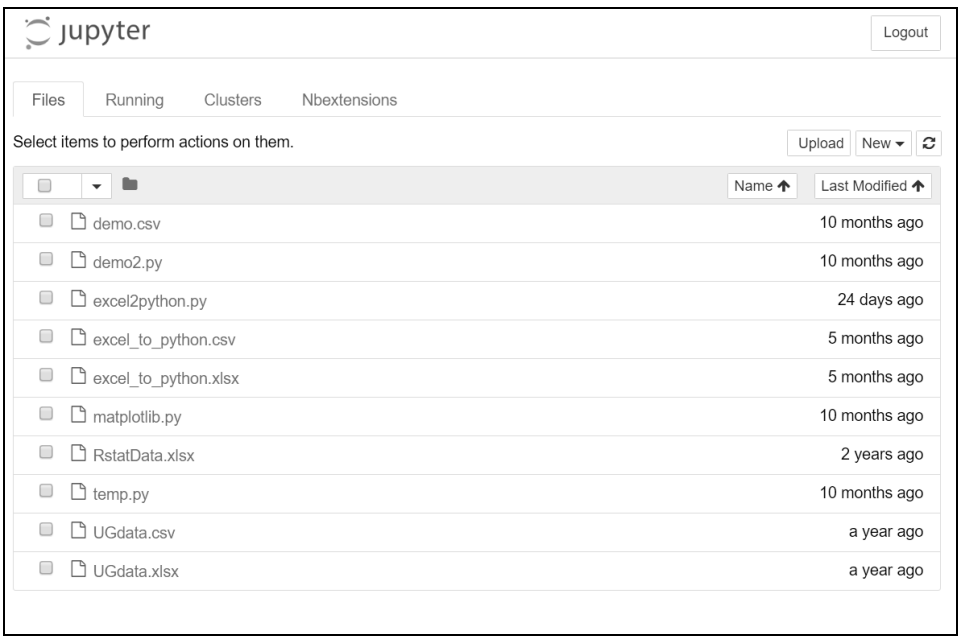
列出当前安装的包：>>> pip list；

列出可升级的包：>>> pip list --outdate；

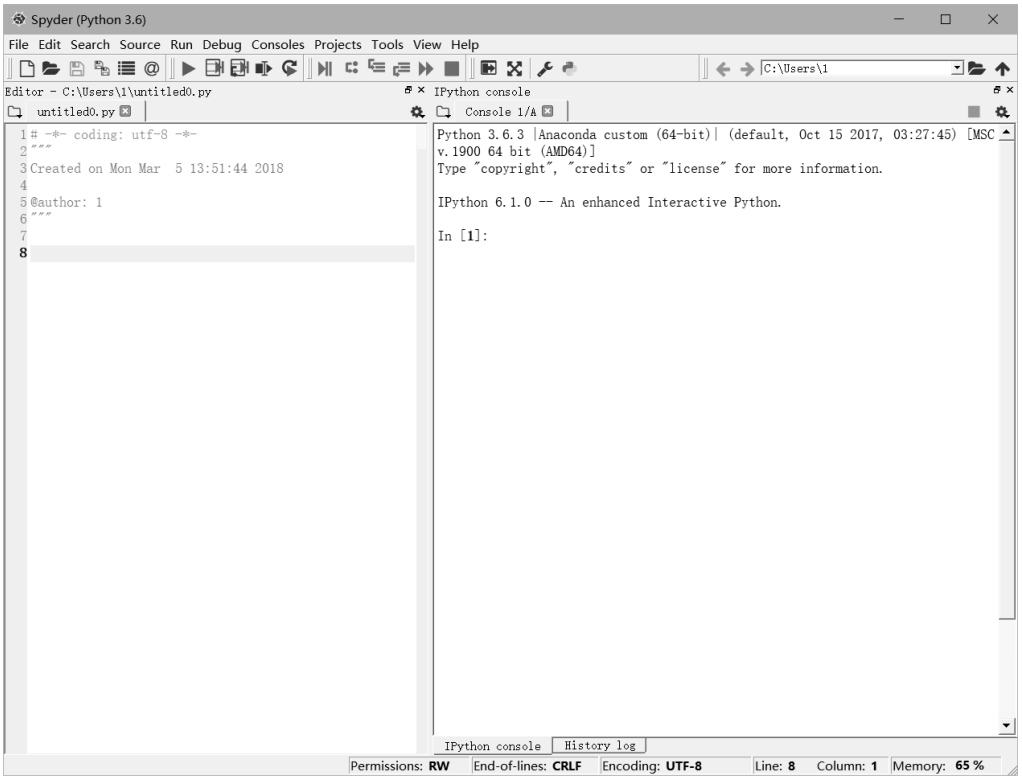
升级一个包：>>> pip install --upgrade jupyterlab；

卸载一个包：>>> pip uninstall jupyterlab。

③ 如果要做基本的数据分析和展示，可执行 Jupyter Notebook。



④ 如果要用 Python 进行大量的数据整理、统计分析、模型建立、程序编写及代码调试，建议使用 Spyder。



9.2 Jupyter 编辑平台

9.2.1 Jupyter Notebook

9.2.1.1 Jupyter Notebook 简介

Jupyter Notebook(此前称为 IPython notebook)是一个交互式编程笔记本，支持运行 40 多种编程语言。Jupyter Notebook 的本质是一个 Web 应用程序，便于创建和共享流程化程序文档，支持实时代码、数学方程、可视化和 markdown。用途包括数据清理和转换、数值模拟、统计建模、数据可视化机器学习等。其特点是用户可以通过电子邮件，Dropbox，GitHub 和 Jupyter Notebook Viewer，将 Jupyter Notebook 分享给其他人。在 Jupyter Notebook 中，代码可以实时生成图像、视频、LaTeX 和 JavaScript。

有时为了能与同行们有效沟通，需要重现整个分析过程，并将说明文字、代码、图表、公式、结论整合在一个文档中。显然，传统的文本编辑工具不能满足这一需求，而 Jupyter Notebook 不仅能在文档中执行代码，还能以网页形式分享。

9.2.1.2 Jupyter Notebook 的使用

建议使用 Anaconda 发行版安装 Python 和 Jupyter，其中包括 Python、Jupyter Notebook、Jupyter Lab，以及用于科学计算和数据科学的其他常用软件包。

如果已经安装了 Jupyter Notebook，要运行笔记本，则在终端(Mac / Linux)或命令提示符(Windows)运行以下命令：Jupyter Notebook。

(1)本地使用

如果安装的是 Anaconda，那么它已包含 Jupyter Notebook，由于 Jupyter 具有网页功能，所以直接打开不易确定当前执行目录，有以下几种在当前目录中打开 Jupyter Notebook 的方法。

① 命令行法。

在 Anaconda Prompt 命令行输入

```
jupyter notebook --notebook-dir= D:\\PyDm1
```

也可以将目录切换为 D:\PyDm1，然后运行 jupyter notebook 命令，如

```
D:\>cd PyDm1
D:\PyDm1>jupyter notebook
```

② Powershell 法。

进入工作目录文件夹(如 D:/PyDm1)→按 Shift 键+单击鼠标右键→打开 Powershell 窗口→在弹出的命令窗口中输入 Jupyter Notebook，如 PS D:\PyDm1\ >Jupyter Notebook。

③ 修改 config.py。

在 CMD(Win+R)中使用 jupyter notebook --generate-config 创建 jupyter_notebook_config.py 文档，在文档中将 c.NotebookApp.notebook_dir = '修改为

```
c.NotebookApp.notebook_dir = 'D:\\PyDm1'
```

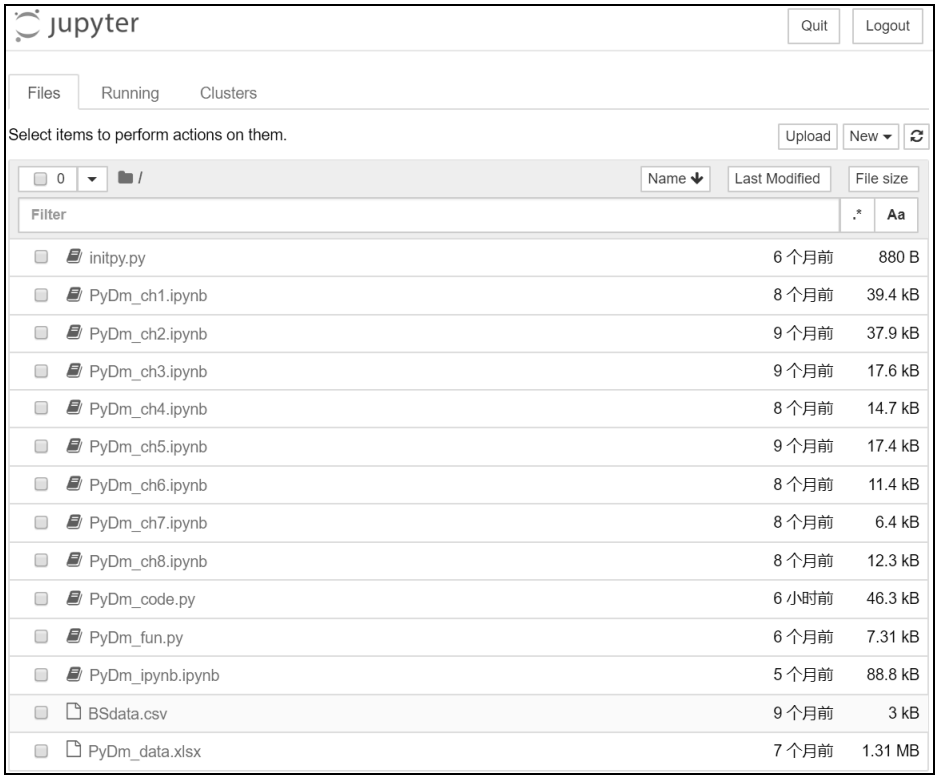
这样，以后每次启动时自动到目录 D:\\PyDm1 下运行。

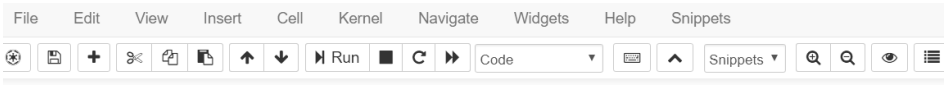
④ 修改 config.json。

打开 Anaconda 安装目录下的 etc 文件，如 C:\Anaconda3\etc\jupyter，本书中安装目录在 C:\Anaconda3，打开 jupyter_notebook_config.json 文件进行如下修改即可：

```
{
"NotebookApp": {
    "nbserver_extensions": {
        "jupyterlab": true,
        "jupyter_nbextensions_configurator": true
    },
    "notebook_dir":"D:\\PyDm1"
  }
}
```

这样，以后每次启动时自动到目录 D:\PyDm1 下运行。





Python数据挖掘方法及应用 PyDm1.0 ——王斌会 王术 2018-12-20

第1章 数据收集与分析软件

第2章 数据挖掘的分析基础

In [1]:
```
## 初始化
%cd "D:\\PyDm1"
import pandas as pd                          #数据分析包
```

```
D:\PyDm1
```

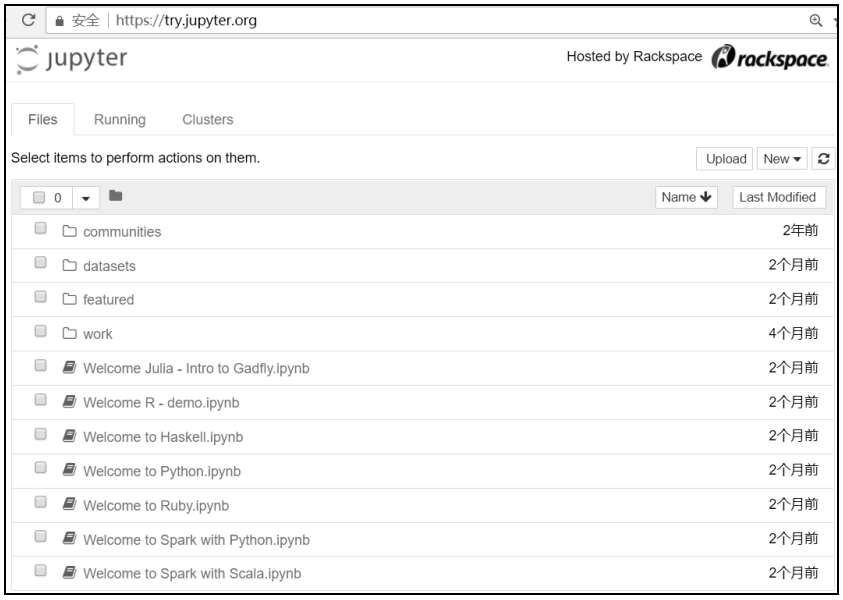
In [2]:
```
BSdata=pd.read_csv("BSdata.csv",encoding='utf-8')  #注意中文格式
BSdata[:9]
```

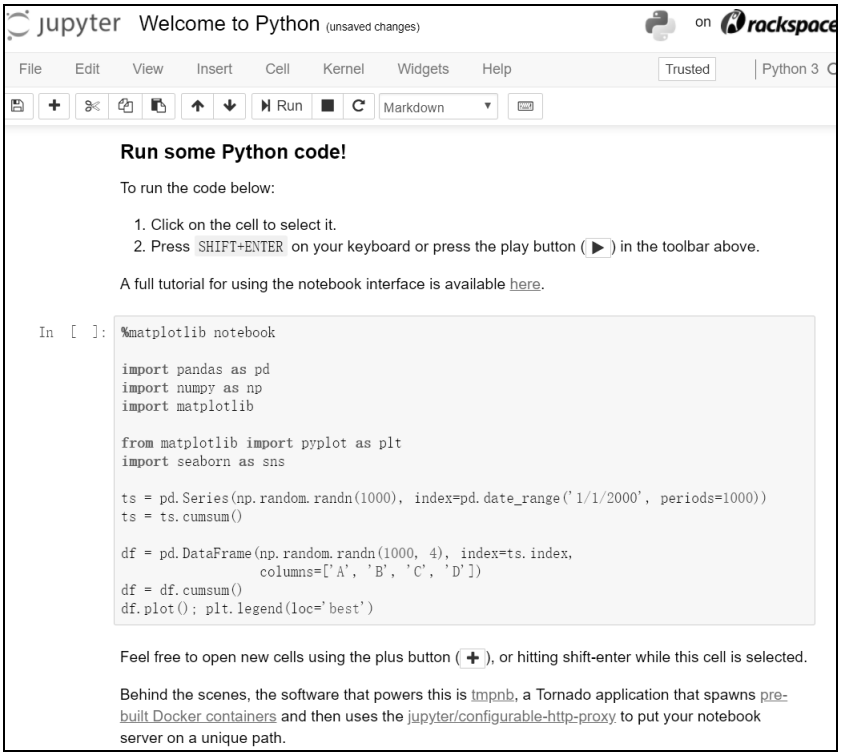
Out[2]:

	学号	性别	身高	体重	支出	开设	课程	软件
0	1510248008	女	167	71	46.0	不清楚	都未学过	No
1	1510229019	男	171	68	10.4	有必要	概率统计	Matlab
2	1512108019	女	175	73	21.0	有必要	统计方法	SPSS
3	1512332010	男	169	74	4.9	有必要	编程技术	Excel
4	1512331015	男	154	55	25.9	有必要	都学习过	Python
5	1516248014	男	183	76	85.6	不必要	编程技术	Excel
6	1516352030	女	169	71	9.1	有必要	编程技术	Excel
7	1516171019	女	166	66	2.5	不必要	都未学过	Excel
8	1516391008	女	165	69	35.6	不必要	都未学过	Excel

(2) 在浏览器中使用

如果不想安装庞大的 Python 和 Jupyter Notebook，而只是简单使用一下，那么可以使用 Jupyter 社区提供的浏览器版 Jupyter Notebook，只要单击【在浏览器中试用】按钮或在网址栏输入 https://jupyter.org/try 即可使用。不过，该版本只包含常用的程序包，一些复杂的程序包还得在本地安装版中使用。





9.2.1.3 Jupyter Notebook 的优点

Jupyter Notebook 的主要优点列举如下。

(1)所见即所得

① 适合进行数据分析。想象如下混乱的场景：你在终端运行程序，可视化结果却显示在另一个窗口中，而包含函数和类的脚本又存放在其他文档中，更可恶的是，你还需另写一份说明文档来解释程序如何执行以及结果如何。此时 Jupyter Notebook“从天而降”，将所有内容收归一处，你是不是顿觉灵台清明，思路更加清晰了呢？

② 支持多语言。Jupyter 支持 40 多种编程语言。如果你习惯使用 R 语言来做数据分析，或者想用学术界常用的 Matlab 和 Mathematica，那么只要安装相对应的核(kernel)即可。

③ 分享便捷。支持以网页的形式分享，GitHub 天然支持 Notebook 展示，也可以通过 nbviewer 分享文档，当然也支持导出成 HTML、Markdown、PDF 等多种格式的文档。

④ 远程运行。在任何地点都可以通过网络连接远程服务器来实现运算。

⑤ 交互式展现。不仅可以输出图片、视频、数学公式，还可以呈现一些互动的可视化内容，比如可以缩放的地图或可以旋转的三维模型。

(2)数学公式编辑

如果你曾做过严肃的学术研究，那么一定对 LaTeX 不陌生，这简直是写科研论文的必备工具，不但能实现严格的文档排版，而且能编辑复杂的数学公式。在 Jupyter Notebook 的 markdown 单元中，也可以使用 LaTeX 的语法来插入数学公式。

在文本行插入数学公式，使用一对 $ 符号，比如质能方程 $E = mc^2$。如果要插入一个数学区块，则使用两对$符号。比如下面的公式表示 z=x/y：

```
$$ z = frac{x}{y} $$
```

关于如何在 notebook 中使用 LaTeX，可进一步参考 A Primer on Using LaTeX in Jupyter Notebooks（http://data-blog.udacity.com/posts/2016/10/latex-primer/）。

(3)幻灯片制作

既然 Jupyter Notebook 擅长展示数据分析的过程，那么除了通过网页形式分享外，当然也可以将其制作成幻灯片的形式。

那么如何用 Jupyter Notebook 制作幻灯片呢？首先在 Notebook 的菜单栏选择 View > Cell Toolbar > Slideshow，这时在文档的每个单元右上角显示了 Slide Type 的选项。通过设置不同的类型，来控制幻灯片的格式，有如下 6 种类型。

- Slide：主页面，通过按左右方向键进行切换。
- Sub-Slide：副页面，通过按上下方向键进行切换。
- Fragment：默认是隐藏的，按空格键或方向键后显示，可实现动态效果。
- Skip：在幻灯片中不显示的单元。
- Notes：作为演讲者的备忘笔记，也不在幻灯片中显示。
- Jupyter Notebook：幻灯片设置。

编写好幻灯片形式的 notebook 以后，如何来演示呢？这时需要使用 nbconvert：

```
jupyter nbconvert notebook.ipynb --to slides --post serve
```

在命令行输入上述代码后，浏览器会自动打开相应的幻灯片。

(4)魔术关键字

魔术关键字（magic keywords），正如其名，是用于控制 notebook 的特殊命令。它们运行在代码单元中，以 % 或 %% 开头，前者控制一行，后者控制整个单元。

比如，要得到代码运行的时间，则可以使用 %timeit；要在文档中显示 matplotlib 包生成的图形，则使用 % matplotlib inline；要做代码调试，则使用 %pdb。注意，这些命令大多是在 Python kernel 中适用的，在其他 kernel 中大多不适用。有许许多多的魔术关键字可以使用，更详细的清单请参考 Built-in magic commands（http://iPython.readthedocs.io/en/stable/interactive/magics.html）。

9.2.2 Jupyter Lab

相信 Python 开发者都对 Jupyter Notebook 这种笔记本式的开发环境非常喜欢。这种基于网页的开发环境不仅允许用户创建和共享含有代码的文档，还可以植入公式、可视化图片和描述性文本等。

然而，所有的东西都不是十全十美的，我们在享受 Jupyter Notebook 的便利的同时，总感觉有种或多或少的缺失感，因为感觉它不太像或压根就不算个 IDE（集成开发环境），

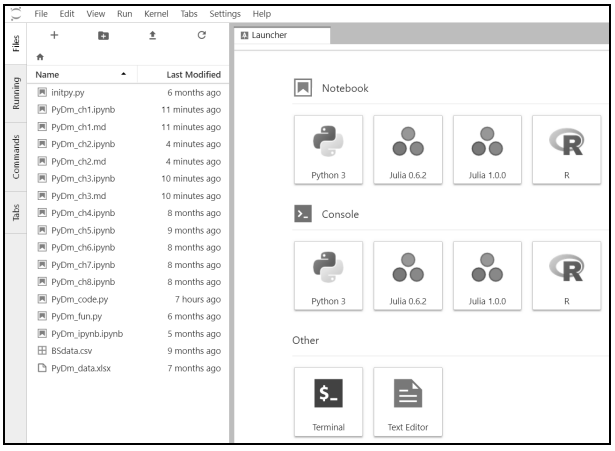
所以看着使用 PyCharm、Spyder 和 Visual Studio For Python 的用户，总有一种莫名的羡慕之感。

令所有开发者为之振奋的好消息是，Jupyter Notebook 的下一代产品 Jupyter Lab 发布了。

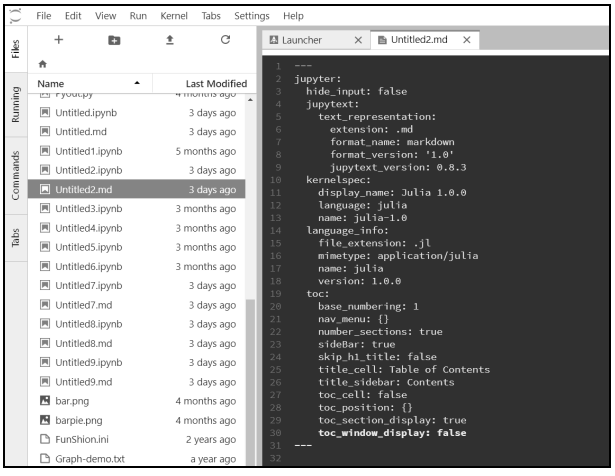
9.2.2.1 Jupyter Lab 的特点

Jupyter Lab 有以下特点。

① 它是一个名副其实的 IDE，且是一个基于网页的 IDE(保留了全部的 notebook 特性)。作者认为，仅凭这一条，Jupyter 项目就是一个飞跃。这个集成环境不仅有 Console，还有 IPython Terminal，所有开发所用到的资源(如图片、代码、文本等)，插件包等，可以在其中运行 Python 和 R 等程序。



② 环境还内置了一个用起来得心应手且功能强大的 markdown 编辑器，这对于编辑程序文档而言十分方便，再也不需要其他的编辑器来撰写 readme 了。与大多数编辑器一样，该编辑器采取对照方式，一边为 markdown 编辑页面，另一边为显示页面。



③ Jupyter Lab 有很多种打开方式，用于打开特定的数据结构和文件格式。比如，要

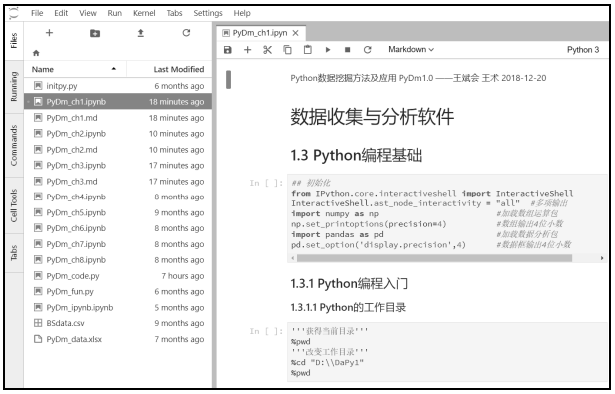
打开一个 csv 文件，除了用 numpy/pandas 就是用 Excel，但 Jupyter Lab 提供了一种表格打开方式，可直接在页面打开这个表形数据，而不是逗号隔开的混乱数据。再如，对于一个 Geo-JSON 文件，如何直观地实现可视化呢？用 Jupyter Lab 以地图形式打开，各个位置就直接显示在 Google Map 中了。

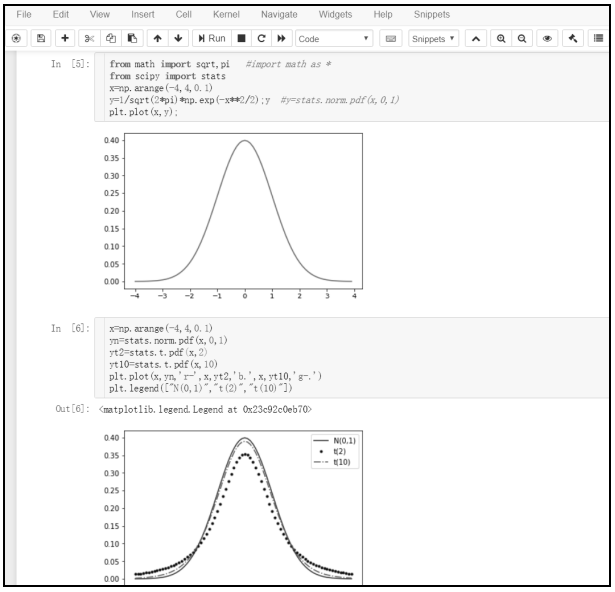
④ Jupyter Lab 扩展了小插件(widget)功能。该功能采纳了其他交互性可视化项目的形式(如 Bokeh)。比如，可以通过滑块(slider)来可视化改变变量值、图形大小、图的分布等。Jupyter Lab 还有很多令人惊喜的功能，这里不再赘述。

9.2.2.2 Jupyter Lab 的使用

如果你安装的是 Anaconda，那么它已包含 Jupyter Notebook 和 Jupyter Lab，由于 Jupyter 具有网页功能，所以直接打开不易确定当期执行目录，可按如下步骤来操作：进入工作目录文件夹(如 D:/PyDm1)→按 Shift 键+单击鼠标右键→在此处打开命令窗口→在弹出的命令窗口中输入 Jupyter Lab，如下图所示。

```
PS D:\PyDm1\>Jupyter  Lab
```





在此就可以像在通常的编程环境中那样来编辑代码和进行数据分析了，操作类似于Jupyter Notebook。

9.2.3 在 Jupyter 中使用 R 语言

Jupyter 的优良性能之一是，可以运行不同语言的内核，下面以运行 R 内核为例来说明。

9.2.3.1 安装 R 内核

(1)通过 Anaconda 安装 R 内核

在命令行执行以下代码即可。

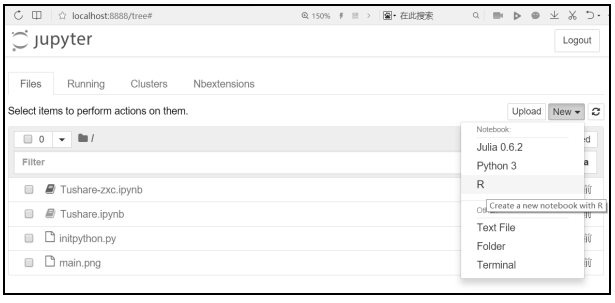
```
>>> conda install -c r r-essentials
```

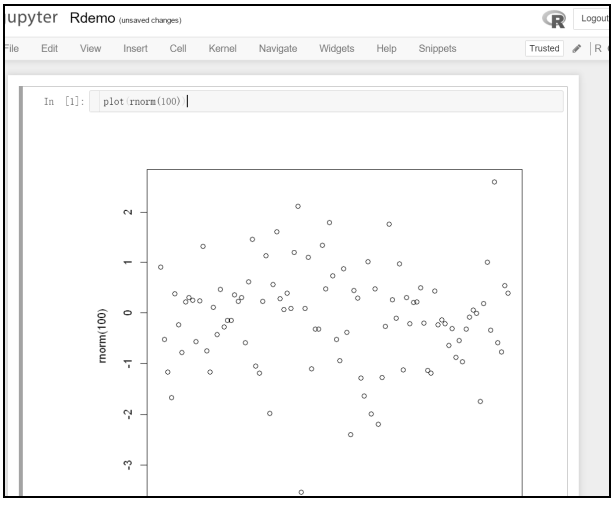
(2)手动安装 R 内核

如果你用的不是 Anaconda，那么过程会稍复杂。首先，从 CRAN 安装 R；然后，启动 R 控制台，运行下面的语句：

```
install.packages(c('repr', 'IRdisplay', 'crayon', 'pbdZMQ', 'devtools'))
devtools::install_github('IRkernel/IRkernel')
IRkernel::installspec()  #to register the kernel in the current R installation
```

(3)创建 R 语言 ipynb





9.2.3.2 同时运行 R 和 Python

要同时运行 R 和 Python，最好的方法是使用 rpy2（电脑中需要有 R 语言的开发环境），在命令行用 pip 可安装 rpy2 包：

```
>>> pip install rpy2
```

然后就可以同时使用两种语言了，变量也可以在二者之间公用：

```
In [1]: %load_ext rpy2.ipython
In [2]: %R require(ggplot2)
Out[2]: array([1], dtype=int32)
In [3]: import pandas as pd
     df = pd.DataFrame({ 'Letter': ['a', 'a', 'a', 'b', 'b', 'b', 'c', 'c', 'c'],
                         'X': [4, 3, 5, 2, 1, 7, 7, 5, 9],
                         'Y': [0, 4, 3, 6, 7, 10, 11, 9, 13],
                         'Z': [1, 2, 3, 1, 2, 3, 1, 2, 3] })
In [4]: %%R -i df
     ggplot(data = df) + geom_point(aes(x = X, y= Y, color = Letter, size = Z))
```

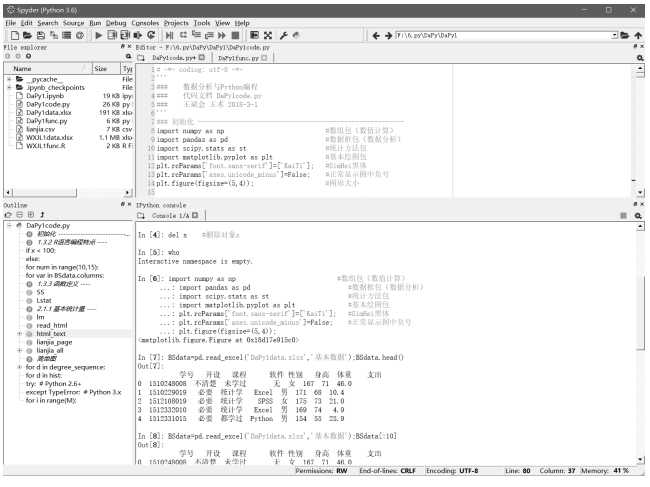
9.3 Spyder 分析平台

9.3.1 Spyder 平台简介

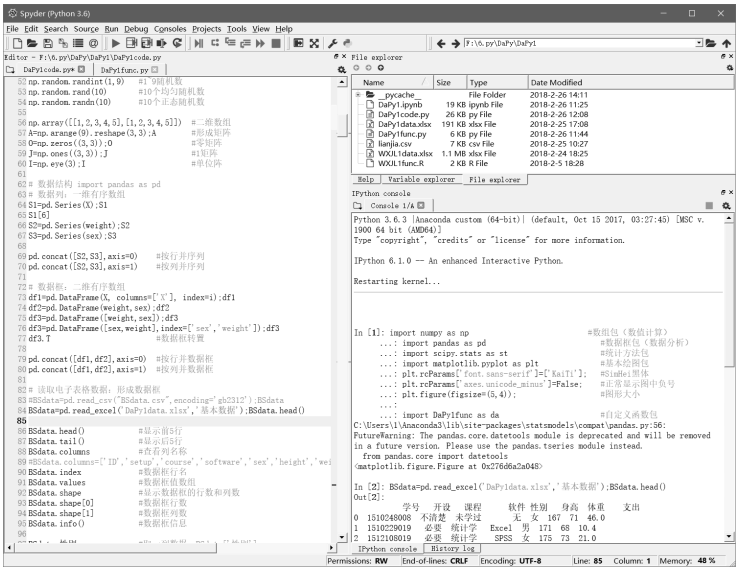
如果要在 Anaconda 中使用 Python 作为数据分析与开发平台，则推荐使用其 Spyder。Spyder 是 Python(x,y)（Python 的一个发行版）的作者为它开发的一个简单的集成开发环境。与其他 Python 开发环境相比，它最大的优点是模仿 Matlab 和 Rstudio 的“工作空间”功能，可以方便地编辑代码和修改数组的值。

如果要进行大量的编程、数据处理和分析工作，可使用 Spyder 编辑器实现类似 Matlab、Rstudio 的开发环境。

下图所示是类似 Matlab 的 Spyder 开发环境。



下图所示是类似 Rstudio 的 Spyder 开发环境。



9.3.2 Spyder 平台使用

关于 Spyder 的详细介绍，参见 Spyder 网站。上面两图就是调整后的 Spyder 界面，实际与 Matlab 和 Rstudio 的编辑器差别不大，但更友好，熟悉 Rstudio 和 Matlab 的用户较易上手。

(1) Spyder 的编辑

Spyder 的界面由许多窗格构成，用户可以根据自己的喜好调整它们的位置和大小。当多个窗格出现在同一个区域时，将以标签页的形式显示。例如，上页图中有 Editor、Object inspector、Variable explorer、File explorer、Console、History log 以及两个显示图像的窗格，在 View 菜单中可以设置是否显示这些窗格。

(2) 功能与技巧

Spyder 的功能比较多，这里仅介绍一些常用的功能和技巧。

默认配置下，Variable explorer 窗格中不显示以大写字母开头的变量，可以单击工具栏中的配置按钮(最后一个按钮)，在菜单中取消 Exclude capitalized references 的选中状态。

在控制台中，可以按 Tab 键进行自动补全。在变量名之后输入“?”，可以在 Object inspector 窗格查看对象的说明文档。此窗格的 Options 菜单中的 Show source 选项可以开启显示函数的源程序。

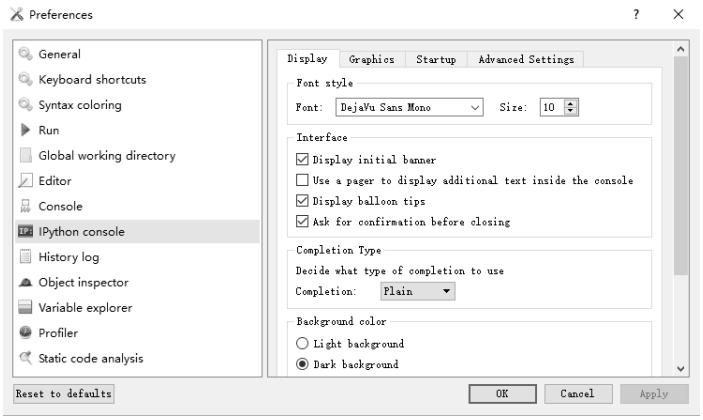
可以通过 Working directory 工具栏修改工作路径，用户程序运行时，将以此工作路径为当前路径。例如，只需要修改工作路径，就可以用同一个程序处理不同文件夹下的数据文件。

在程序编辑窗口中按住 Ctrl 键的同时单击变量名、函数名、类名或模块名，可以快速跳转到定义位置。如果是在别的程序文件中定义的，则将打开此文件。在学习一个新

模块的用法时，经常需要查看模块中的某个函数或类是如何实现的，使用此功能可以快速查看和分析各个模块的源程序。

(3) Spyder 的配置

基本的配置都在 Tool→Perference 中。



附录 A　本书的学习网站

为方便读者使用本书，我们建立了学习博客(http://blog.leanote.com/DaPy)，书中的例子数据和习题数据都可直接在网上下载使用。

附录 B　书中的例子数据

本书的数据全都保存于文档 PyDm_data.xlsx 的表单中，可在其中下载。

附录 C　书中自定义函数

1．为便于大家学习本书及使用 Python 进行数据分析，书中自定义了一些 Python 函数辅助进行数据分析，下面列出这些函数所在章节及其用途。

本书包的名称为 PyDm_fun，可在其中下载。

为便于读者使用这些函数，下面提供获得函数和包的途径。

在使用 Python 前，最好在本地建立一个目录，这样你的所有数据、代码及计算结果都可保存在该目录下，方便操作。这里假设建立的目录是 D:\PyDm，然后将本书所有自编函数形成一个 Python 文档 PyDm_fun.py，读者可加载调用。

2．自定义函数包的安装与使用

直接调用函数：

(1) 安装自定义模块：将 PyDm_fun.py 文档复制到当前工作目录 D:\PyDm 下。

(2) 加载自定义模块：from PyDm_fun import *。

(3) 自定义函数调用：mcor_test（X）#使用相关系数矩阵检验函数。

更正规的调用方法：

(1) 安装自定义模块：将 PyDm_fun.py 文档复制到当前工作目录 D:\PyDm 下。

(2) 加载自定义模块：import PyDm_fun as dm。

(3) 自定义函数调用：dm.mcor_test（X）#使用相关系数矩阵检验函数。

3．书中自定义函数的源代码

(1) 相关阵检验函数

```
def mcor_test(X):  #相关系数矩阵检验
    p=X.shape[1];p
```

```
    sp=np.ones([p, p]);sp
    for i in range(0,p):
        for j in range(i,p):
            sp[i,j]=st.pearsonr(X.iloc[:,i],X.iloc[:,j])[1]
            sp[j,i]=st.pearsonr(X.iloc[:,i],X.iloc[:,j])[0]
    R=pd.DataFrame(sp,index=X.columns,columns=X.columns)
    print(round(R,4))
    print("\n 下三角为相关系数，上三角为概率")
```

(2)主成分分析函数

```
def PCrank(X,m=2):  #主成分评价函数
    from sklearn.decomposition import PCA
    Z=(X-X.mean())/X.std()
    p=Z.shape[1]
    pca = PCA(n_components=p).fit(Z)
    Vi=pca.explained_variance_;Vi
    Wi=pca.explained_variance_ratio_;Wi
    Vars=pd.DataFrame({'Variances':Vi},index=X.columns);Vars
    Vars['Explained']=Wi*100;Vars
    Vars['Cumulative']=np.cumsum(Wi)*100;
    print("\n 方差贡献:\n",round(Vars,4))
    Compi=['Comp%d' %(i+1) for i in range(m)]
    loadings=pd.DataFrame(pca.components_[:m].T,columns=Compi,index=X.columns);
    print("\n 主成分负荷:\n",round(loadings,4))
    scores=pd.DataFrame(pca.fit_transform(Z)).iloc[:,:m];
    scores.index=X.index; scores.columns=Compi;scores
    scores['Comp']=scores.dot(Wi[:m]);scores
    scores['Rank']=scores.Comp.rank(ascending=False);scores
    print('\n 综合得分与排名:\n',round(scores,4))
    plt.plot(scores.Comp1,scores.Comp2,'.');
    for i in range(Z.shape[0]):
        plt.text(scores.Comp1[i],scores.Comp2[i],X.index[i])
    plt.hlines(0,scores.Comp1.min(),scores.Comp1.max(),linestyles='dotted')
    plt.vlinZes(0,scores.Comp2.min(),scores.Comp2.max(),linestyles='dotted')
```

(3)网络爬虫函数

```
import requests
from bs4 import BeautifulSoup
def read_html(url,encoding='utf-8'):   #获取 html 文档
    response = requests.get(url)
    response.encoding = encoding
    return response.text
def html_text(info,word):              #按关键词解析文本
    return([w.get_text() for w in info.select(word)])
```

(4)链家二手房信息收集与处理函数

```
def lianjia_page(Soup):                #单个网页信息
    lianjia=pd.DataFrame()
    lianjia['房屋信息']=html_text(Soup,'.houseInfo')
```

```
    lianjia['房屋价格']=html_text(Soup,'.totalPrice span')
    lianjia['房屋位置']=html_text(Soup,'.positionInfo a')
    lianjia['房屋单价']=html_text(Soup,'.unitPrice span')
    return(lianjia)

def lianjia_all(url,long):#所有网页信息
    houseinfo=pd.DataFrame()
    for i in range(long):
        web=read_html(url+str(i))
        soup=BeautifulSoup(web,'lxml')
        pages=lianjia_page(soup)
        houseinfo=pd.concat([houseinfo,pages])
    return(houseinfo)

def list_replace(content,old,new): #清除信息中的空格
    return [content[i].replace(old,new) for i in range(len(content))]

def list_split(content,separator):#分解信息
    new_list=[]
    for i in range(len(content)):
        new_list.append(list(filter(None,content[i].split(separator))))
    return new_list
```

(5)文献计量分析函数

```
def find_words(content,pattern):#寻找关键词
    return [content[i] for i in range(len(content)) if (pattern in content[i]) == True]

def search_university(content,pattern): #查找单位
  return len([find_words(content[i],pattern) for i in range(len(content)) if
find_words(content[i],pattern) != []])
```

(6)共现矩阵的计算

```
def occurence(data,document):  #定义共现矩阵
    empty1=[];empty2=[];empty3=[]
    for a in data:
        for b in data:
            count = 0
            for x in document:
                if  [a in i for i in x].count(True) >0 and [b in i for
                                i in x].count(True) >0:
                        count += 1
            empty1.append(a);empty2.append(b);empty3.append(count)
    df=pd.DataFrame({'from':empty1,'to':empty2,'weight':empty3})
    G=nx.from_pandas_edgelist(df, 'from', 'to', 'weight')
    return (nx.to_pandas_adjacency(G, dtype=int))
```

参 考 文 献

[1] 王斌会，王术. Python 数据分析基础教程. 北京：电子工业出版社，2018.

[2] 王斌会. 数据统计分析及 R 语言编程. 2 版. 北京：北京大学出版社，2017.

[3] 王斌会. 计量经济学模型及 R 语言应用. 北京：北京大学出版社，2015.

[4] 王斌会. 多元统计分析及 R 语言建模. 4 版. 广州：暨南大学出版社，2016.

[5] 谢贤芬，王斌会. Excel 在经济管理数据分析中的应用. 北京：北京大学出版社，2015.

[6] 吴国富，安万福，刘景海. 实用数据分析方法. 北京：中国统计出版社，1992.

[7] 唐启义，冯明光. 实用统计分析及其 DPS 数据处理系统. 北京：科学出版社，2002.

[8] Wes McKinney. 利用 Python 进行数据分析. 唐学韬，等译. 北京：机械工业出版社，2014.

[9] 张良均，王路，谭立云，苏剑林. Python 数据分析与挖掘实战. 北京：机械工业出版社，2015.

[10] Fabio Nelli. Python 数据分析实战. 杜春晓，译. 北京：人民邮电出版社，2016.

[11] 吴喜之. Python——统计人的视角. 北京：中国人民大学出版社，2018.

[12] Python 数据分析学习博客：http://blog.leanote.com/DaPy.

反侵权盗版声明